Statistical Neuropsychology
III Methodology

Zach Wolpe

05 Nov 2021

Abstract

An overview of the statistical mechanics frequently utilized in computational neuropsychology.

Contents

1 Foundations of Reinforcement Learning

Reinforcement learning (RL) offers an intuitive framework to formalize any stochastic sequential decision making process [15]. A consequence of this broad & ubiquitous utility has resulted in RL being studied from a plethora of vantage points. Engineers study optimal control; mathematicians and economists study operations research and bounded rationality; Neuroscientist and psychologist study reward systems and classical/operant conditioning; and of course, machine learning partitioners study reinforcement learning. Each of these disciplines can be considered special cases of the broader RL/markovian framework [16].

There are 4 characteristics that make RL distinct from other machine learning paradigms:

1. The models are unsupervised but rewards are given to proxy performance.
2. Feedback is delayed.
3. Data is sequential and not i.i.d.
4. Actions affect subsequent data received [15].

RL is primarily concerned with describing a sequence of choices that some *agent* takes in some *environment*. The agent could describe a robot, or car, or people in an economy or a trading bot - anything that requires sequential decision making [15].

Here we discuss one specific instance of RL - too illustrate it's practicalities - as well as the theory in abstract. Consider a simple instance of a robot that wishes to navigate a 2-dimensional maze.

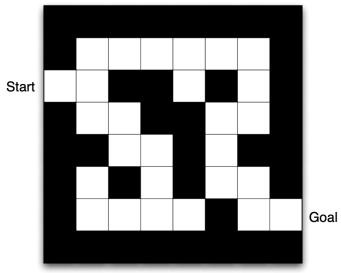

Figure 1: A simple state space environment [15].

Given the maze, an agent placed in some location $State : s$ aims to move to the terminal state by taking some sequence of *actions* drawn from some policy π ($a_t \sim \pi$). More generally, some agent wishes to achieve a specified objective by taking some set of actions from the space of all possible actions A & thus traversing through the state space S. It is important to note the universality of this state space formalization - applicable to any sequential decision making problem.

Let a_t denote the action taken at time t. We then quantify the quality of the actions taken by an agent as the total *reward* attained over the agents lifetime, defined as some metric of success describing the agents objective [17]. Let $G = \sum_{t=1}^{\infty} r_t$, where r_t is the reward received at time t [17]. Reward is discounted for mathematical convenience (avoiding infinite yields in the limit) [17] as well as logical consistency (representing the time value of returns, capturing uncertainty

of future yield and maintaining consistency with empirical evidence of near term preference) [15] thus $G = \sum_{t=1}^{\infty} \gamma^t r_t < \infty$ where $\gamma \in [0, 1]$ Our agent objective is to act such that we maximise the expected return:

$$\underset{\pi}{\text{argmax}}\, G = \sum_{t=1}^{\infty} \gamma^t r_t^{\pi}$$

.

1.1 Definitions

RL models are a special case of Markov Processes [4]. Markov models are stochastic processes that model pseudo-random dynamic systems [4]. A key tenant of Markov chains, known as the Markov property, is that future states depend only on the current state and not prior states [15]. A pragmatic quality as it makes otherwise intractable models computable [4]. We define the following probabilities:

$$\begin{aligned} s_t \in S &: \text{an instance from the } \textit{state} \text{ space.} \\ a_t \in A &: \text{an instance from the } \textit{action} \text{ space.} \\ T^a_{ss^{'}} = p(s^{'}|s, a) &: \text{state action transition probabilities.} \\ r_t \sim R(s_{t+1}, a_t, s_t) &: \text{a sample from possible rewards.} \\ \pi(a|s) = p(a|s) &: \text{the policy under which an agent acts [15].} \end{aligned}$$

where t denotes the time period; $T^a_{ss^{'}}$ describes the probability of traversing from state s to state $s^{'}$ when employing action a; and $\pi(a|s)$ - the *policy* - describes the probability distribution over the actions given the current state [17].

1.2 Deriving the value function

We aim to select a policy that maximised all future discounted rewards G. Let us denote the value of a state s when following a particular policy π as $v_\pi(s)$. If the state values are known, our agent can simply act greedily with respect to the state values - opting for the policy that maximizes G [16]. In the example given in figure 2 this corresponds to moving towards the terminal state [15]. The policy determines the expected reward in each state thus we define a state value by [15]:

$$V^{\pi}(s) = \mathbb{E}[\sum_{t=1}^{\infty} \gamma^t r_t | s_t, a_t \sim \pi]$$

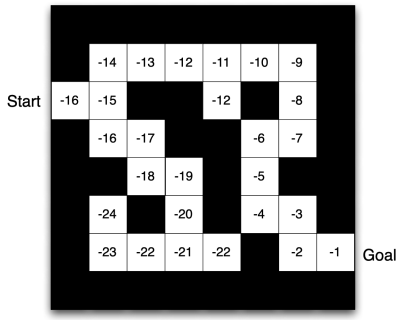

Figure 2: An example of a state space with known true state values. If the object is to exit the maze, the state values would correspond inversely with their distance to the terminal state. An agent could then act greedily in accordance with these state values to move to the terminal state in an optimal fashion [15].

It is paramount to note that the value function can be compartmentalized into the immediate reward and all future rewards, shown here [15]:

$$V^{\pi}(s) = \mathbb{E}[\sum_{t=1}^{\infty} \gamma^t r_t | s_t, \pi]$$

$$V^{\pi}(s) = \mathbb{E}[r_1 | s_t, \pi] + \mathbb{E}[\sum_{t=2}^{\infty} \gamma^t r_t | s_t, \pi]$$

Various methods sample n rewards and then approximate the remaining value function [16]. Recall that expectation is given as the weighted sum of all possible options, it can then be shown [15]:

$$\begin{aligned}
\mathbb{E}[r_1 | s_t, \pi] &= \mathbb{E}[\sum_{s_{t+1}} p(s_{t+1} | s_t, a_t) R(s_{t+1}, a_t, s_t)] \\
&= \sum_{a} p(a|s_t)[\sum_{s_t+1} p(s_{t+1} | s_t, a_t) R(s_{t+1}, a_t, s_t)] \\
&= \sum_{a} \pi(a|s_t)[\sum_{s_{t+1}} T^{a_t}_{s_t, s_{t+1}} R(s_{t+1}, a_t, s_t)]
\end{aligned}$$

And similarly:

$$\mathbb{E}[V^{\pi}(s_t) | s_t, \pi] = \sum_{a} \pi(a|s_t)[\sum_{s_{t+1}} T^{a}_{s_t, s_{t+1}} V^{\pi}(s_{t+1})]$$

Thus, we arrive at the paramount Bellman Equation [15]:

$$V^{\pi}(s_t) = \sum_{a} \pi(a|s_t) \left[\sum_{s`} T^{a}_{s, s`} [R(s`, a, s) + V^{\pi}(s`)] \right]$$

Which can be interpreted as all future rewards in state s are equivalent to the immediate reward plus all future rewards from the next state $s^{'}$ [15]. This quantity is averaged over all actions. If we instead assume a single *action*, we derive the $state, action$ values [17]:

$$\begin{aligned} Q(s,a) &= \sum_{s^{'}} T^{a}_{s,s^{'}}[R(s^{'},a,s) + V^{\pi}(s^{'}) \\ &= \mathbb{E}\left[\sum_{t=1}^{\infty} r_t | s, a\right] \end{aligned}$$

Note, the *state* values are the average over $state - action$ values $V(s) = \sum_a \pi(a|s)Q(s,a)$.

1.3 Solutions to the Bellman equation

1.3.1 Model-based RL

Too evaluate a policy we need to solve the Bellman expectation equation [17]:

$$V^{\pi}(s_t) = \sum_a \pi(a|s_t)\left[\sum_{s^{'}} T^{a}_{s,s^{'}}[R(s^{'},a,s) + V^{\pi}(s^{'})]\right]$$

In theory, be solved analytically. If we express the Bellman equation in matrix notation:

$$\begin{aligned} \mathbf{v}^{\pi} &= \mathbf{R}^{\pi} + \mathbf{T}^{\pi}\mathbf{v}_{\pi} \\ \implies \mathbf{v}^{\pi} &= (\mathbf{I} - \mathbf{T}^{\pi})^{-\mathbf{1}}\mathbf{R}^{\pi} \end{aligned}$$

This can also be posed as an update equations, for instances where the linear solutions is intractable [17]:

$$V^{k+1}(s) = \sum_a \pi(a|s_t)\left[\sum_{s^{'}} T^{a}_{s,s^{'}}[R(s^{'},a,s) + V^{k}(s^{'})]\right]$$

One can then update the policy by selecting the value that maximises the expected return, that is:

$$\pi(a|s) = \begin{cases} 1 \; if \; a = \underset{a}{argmax} \sum_{s^{'}} T^{a}_{ss^{'}}[R^{a}_{ss^{'}} + V^{\pi}(s^{'})] \\ 0 \; otherwise \end{cases}$$

By these update rules one can iteratively search the solution space to arrive at the optimal policy.

Note the solutions here require an accurate model of the environment - knowledge of transition dynamics T and reward distribution R [15]. This is the domain of ***model-based RL***, an alternative ***model-free RL*** aims to estimate V and Q directly from the data [17].

1.3.2 Model-free RL

If we instead take actions in the environment and observe the results that are returned, we're effectively sampling the state space [15]. Once our state space; action space and objective are defined we need to solve for the state values to act in accordance with the objective - that is, to find the optimal *policy* [15].

Recall that in the absence of a world model (knowledge transition dynamics and reward distributions) the exact solution would require permuting over all possible sequences - which readily becomes intractable, illustrated in figure 3 [17]. We instead employ approximation methods and dynamic programming to iteratively search the solution space [17].

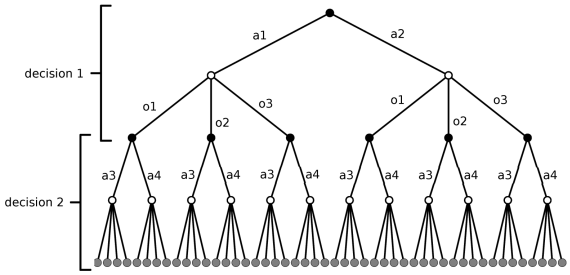

Figure 3: An illustration of the branching factor of exhaustive search [15].

It is important to intuit the data generating process utilized by Reinforcement Learning - a key dissimilarity from other machine learning techniques [16]. The data are the rewards generated by interacting with the environment. Thus exhaustive search (figure 3 is to compute all possible trajectories across the action space and simply choose the action that maximises aggregated reward. This if of course idealistic and impractical all non-trivial problems [17].

Like any statistical method we aim to interpolate data. Although exhaustive search may be intractable, various dimensions of sampling the state/action space can result in adequately estimating state values - thus learning the optimal policy [15].

1.3.3 Dynamic programming

A plethora of reinforcement learning methods rely on dynamic programming to iteratively search the solution space [17]. Dynamic programming is applicable as the same states are crossed for n policies, thus allowing for a method of caching and reusing state values [16]. In the context of RL, dynamic programming is implemented by sweeping over all possible states and performing one step look ahead (computing the result for taking a single action) - and using this feedback to update the model [16]. Subsequent iterations then follow some greedy policy (to avoid excessive sweeping computation) to iteratively update value estimates (V^{π} - visualized in figure 4. Whilst many variants exist, the paramount principles are:

- Modularising the problem to avoid redundant computation (shared states value estimates) and then,
- Selectively sweeping over the state space to update value estimates [15].

Dynamic programming can us be considered a breadth first search approach [4].

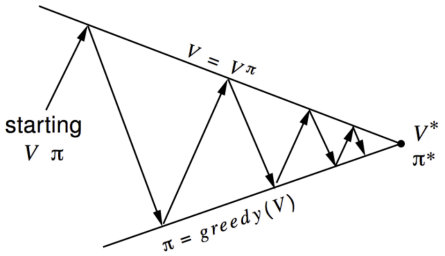

Figure 4: A graphical representation of dynamic programming - iteratively improving value estimates whilst following the current optimal policy greedily. Guaranteed to converge to the optimal state values and policy (v^* and π^*) in the limit [16].

1.4 Monte Carlo learning

On the other end of the spectrum, we may conduct purely depth first search via Monte Carlo sampling [16]. When employing Monte Carlo, we role out a single policy until reaching the terminal state and then approximate the current state value by the weighted average of the yielded returns [16], formally:

$$V(s) = \frac{1}{N} \sum_i \left\{ \sum_{t=1}^{T} \gamma^t r_t^i | s_0 = s \right\}$$

where $r_t^i \sim R(s, a, s`)$ is a sample reward.

1.5 Temporal Difference (TD) learning

On the the other extreme whereby value estimates are updates immediately after every step - sampling the reward returned by taking action a in state s and updating $V^\pi(a)$. This method is referred to as Temporal Difference Learning (TD-learning) or bootstrapping [16]. Given the Bellman equation:

$$V(s) = \sum_a \pi(a|s) \left[\sum_{s`} T_{ss`}^a [R(s`, a, s) + V(s`)] \right]$$

Under the assumption that we have not yet converged to the optimal policy, the margin change in $V(s)$ is defined as:

$$dV(s) = -V(s) + \sum_a \pi(a|s) \left[\sum_{s`} T_{ss`}^a [R(s`, a, s) + V(s`)] \right]$$

which results in the update rule [16]:

$$V^{i+1}(s) = V^i(s) + dV(s)$$

TD learning samples from the respective distributions:

$$\begin{aligned} a_t &\sim \pi(a|s_t) \\ s_{t+1} &\sim T^{a_t}_{s_t,s_{t+1}} \\ r_t &\sim R(s_{t+1}, a_t, s_t) \end{aligned}$$

Letting δ_t denote our sample of $dV(s)$:

$$\begin{aligned} \delta_t &= -V_{t-1}(s_t) + r_t + V_{t-1}(s_{t+1}) \\ \implies V_t(s) &= V_{t-1}(s) + \alpha\delta_t \end{aligned}$$

It is straightforward to see how this can be interpreted as reward prediction error, updating our belief proportionally to the discrepancy between our current believe and feedback from the environment [16].

It is also intuitive to see the bias-variance trade off when sampling the solution space: TD learning updates value estimates with a single sample (return R_t) whilst Monte Carlo learning runs a policy to completion utilising many returns $\sum_{t=1}^{T} \gamma^t R_t$. Whilst Monte Carlo is then less biased - reliant on a greater sample from the state space - it exhibits vastly greater variance as each individual r_t is a sample reward with associated stochastic fluctuations [16].

Finally, all of these techniques can be consolidated by means of TD(λ) update rules - where λ dictates the number of samples required to perform an update - accompanied by some policy update rule [16]. Figure 5 exhibits the state of possible search algorithms captured in the Markovian RL framework [17].

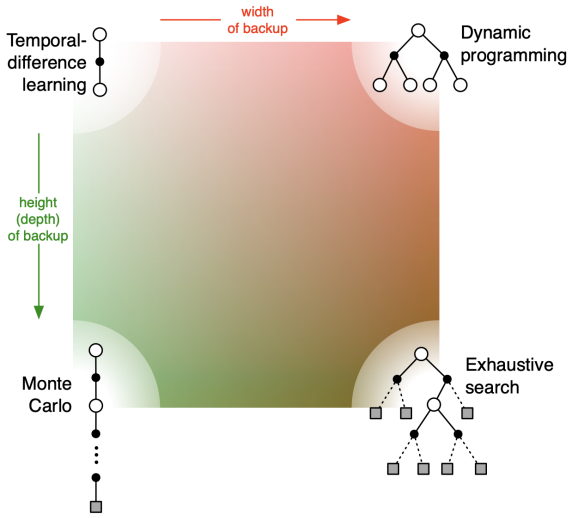

Figure 5: A slice of the space of reinforcement learning methods [16].

The manner in which we interpolate between policies is governed by the infamous ***exploration-exploitation*** trade-off [17]. It can be interpreted a driving function that determines the transitions

between policies. An exploration parameter is often of keen interest in computational psychiatry as it pertains to an individuals risk aversion.

There are copious extensions and variations too these methods, however they are beyond our requirements. The methods discussed here can be considered look up tables, many applications of interest instead attempt to fit some parametric function/distribution to the data (states, actions & additional attributes) to approximate state values) [15].

It's worth noting that different variants of RL focus on learning the optimal policy directly (policy gradient methods) or solving for state values to inform the optimal policy [17]. The latter is more interesting in our case as in neuropsychology we're concerned with understand the driving forces behind actions - replying on RL as an approximation of cognition & thus deriving interpretable parameters.

2 Motivation behind Computational Psychiatry

Many, if not all, modern scientific endeavours have become increasingly data driven - capitalising on both the new found empirical evidence and the exponentially cheapening cost to compute [9]. Recent advancements in computational neuroscience, psychiatry and psychology, however, transcend this pure practical utilisation of large data & available compute and have begun to utilise computational methodologies in designing theoretical models [9]. That is, we have now entered a new era of neuroscience and psychiatry poses the brain as a computational system, taking theory from our knowledge of computational systems to infer characteristics of the mind [1].

One compelling example is the *Bayesian brain* hypothesis, which poses the brain as a Bayesian system that acts to collect model evidence [10]. We are constantly inundated with sensory information and are tasked with distilling & inferring reasonable responses to these signals. There is growing evidence to support the idea that neural-computation is analogous to Bayesian optimisation in that we act in accordance with Bayesian principles [13].

Information about uncertainty and prior beliefs are iteratively updated in light of new information - guiding our perception and sensorimotor control [10]. It then naturally follows that the brain may be representing sensory information probabilistically - assuming probability distributions over the physical world [10]. In the way, brain function can be posed as inferential processes - combining prior beliefs with a generative (predictive) model to infer the causes of sensations [13].

A fascinating sub-field has emerged where neuropsychological deficits, psychiatric disorders and pathological mental illnesses are poses as improperly fitting model evidence to one's generative model by acting on aberrant priors - those with a poor fit to the real world [13]. That is to suggest these mental defects as false inferences derived by performing Bayesian optimisation over improper priors [13]. Applicable to a vast array of systems from visual neglect through hallucinations to autism, the utility of these methods surpass simple intellectual gymnastics, but offer plausible, theoretically rich, and implementable frameworks to generate (and test) hypothesis and aid in deriving novel solutions to exceedingly complex problems [10].

The key theoretical premise of this link between biology and computation is based on the idea that neurological activity is designed to compute estimates about the physical world [7].

2.1 Forms of neural computation

If when then rely on the assumption that neurological activity is fundamentally analogous to computation, under which conditions can we theorize mental illness? One useful categorisation is between errors arriving in *inference* or in *learning* [7]. Psychosis or cognitive distortions are indicative of poor inference, as an individuals is incapable of accurately assessing reality [2]. Additional neurological defects may arise through learning. The manner in which an individual's brain assimilates information is a function of the information it is exposed to; which in turn can change how future information is processed [7]. As such, exposure to certain information - such as trauma, ill-parenting or adverse life-events - may substantially distort the way in which an individual processes information [7].

2.2 Computational approaches to modeling neuroscience

Whilst nuances exist, the computational approaches taken to model neurological activity can be split into 3 categories [8]:

- Dynamic Systems
- Inference
- Learning

2.2.1 Dynamic Systems

Often modelled by a series of differential equations, dynamical systems can be used to describe how *nodes* on some *graph* interact to produce certain behaviour [2]. This may arise when modeling the anatomy of the brain - understanding the relationship between action potentials - but can be extended to any level of abstractions [2]. Dynamic systems can captures how an individuals symptoms affect their condition (for example, how a chronically lonely individual may act in anti-social manner perpetuating their condition) [7]. Effectively modelling any set of interactions & transition dynamics.

2.2.2 Inferential Models

Inference in this context is are concerned with learning the generative process behind certain actions/behaviours [7]. That is, given some sensory input, can we infer the underlying latent structure that produces some given output [2].

2.2.3 Learning

Finally, learning mechanisms are concerned with how information changes future behaviour - how internal state, priors, assumptions and habits are updating in light of additional data [7]. Consider a person who wishes to play chess. One likely approach taken would be to learn the rules of the game & then attempt to simulate (consider different plays) at each step of a given game in order to predict a good outcome - acting in a manner to win the game. If the person could consider all possible game trajectories, the optimal policy could be chosen - note this is comparable to exhaustive search in the Reinforcement Learning literature [7]. If, instead, the individual relies on pattern matching, habitual behaviour and past experience to estimate the next best move, we can conclude the individual is building a *model* of the *environment* and acting in accordance with this model's projections [7]. In the psychological literature refers to these learning mechanisms as *model-based* or *goal-driving* learning.

A seminal finding in contemporary neuroscience is mechanism under which conditioning - a type of learning where a stimulus is associated with some outcome - pertains to dopamine neurons [14]. The formative findings show that the release of dopamine (and thus the reward received) does not directly correspond to the actual reward received in reality, but rather the difference between the rewarded and the ***expected*** reward [14]. As detailed in figure 6 extensive experimentation has been documented to show that the release of dopamine in the midbrain is dependent on expected (rather than absolute) reward [14]. This is show by conditioning participants to expect some reward, in this example a beverage or sweet snack, after some sound or visual cue (the CS or conditioned stimulus). Before learning the association, and when no stimulus is provided, the the onset of the reward R results in a spike in dopamine (seen in the top panel of figure 6). Individuals are not expecting this rewards and thus are pleasantly surprised; that is, there is a great positive discrepancy between the expected reward and actual reward [14]. The middle and bottom panel of figure 6 show the dopamine response after learning the association. Not in both instances the

stimuli alone is sufficient to trigger the dopamine release, which precedes the actual reward. After the stimuli, the participants expectations have been shifted to expect a reward, so when the actual reward is received dopamine remains relatively stable. Interestingly, if the reward is then omitted, a notable drop in dopamine occurs - indicative of this misalignment with expectations [14]. This expectation-reward discrepancy is known as ***RPE: reward prediction error***.

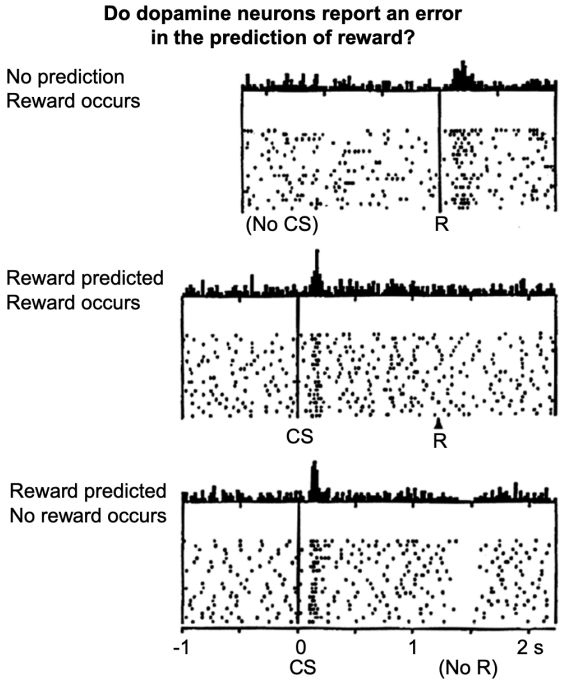

Figure 6: Seminal work detailing the relationship between releases of dopamine in the midbrain and reward prediction error (RPE) [14].

This important finding is the birthplace of utilizing reinforcement learning as a plausible framework for human learning - allowing parameter estimates to update in accordance with reward prediction errors [1]. This is to be discussed at length in later chapters.

2.3 Psychiatric analysis through computational models

This mathematical theory driven approach to computational psychiatry is distinct from pure machine learning & statistical endeavours, not by the means under which the models are fit, but by the a priori decisions made in selecting the model architecture [1]. The data driven approach avoids theoretical assumptions, relying purely on statistical inference to reveal relationships in the (physiological and psychological) data; whilst the theoretical approach utilises the rich mathematical and statistical literature to define mathematically precise hypothesis about the data, by implementing the following procedure [2]:

- ***Model Specification***: a statistical formulation about the (latent) data generating process.

- ***Model Estimation***: Fitting parameters to the observed data.
- ***Model Comparison***: Finding an optimal balance between model complexity and explainability to fit the parsimonious model.
- ***Model Testing***: Assessing how well the optimal model recapitulates the observed data [2].

This theoretical model formulation can be illustrated by RPE example provided in figure 6. The $Q - Learning$ update equation can adequately capture the learning process:

$$Q_{t+1}(a) = Q_t(a) + \alpha \left[R_t - Q_t(a)\right]$$

Which intuitively tells us the amount which the state value is update is proportional to the RPE:

$$Q_{t+1}(a) = Q_t(a) + \alpha \left[RPE\right].$$

Thus the parameter α in this example is indicative of the individuals learning rate [8]. Importantly, the parameters have interpretable theoretical groundings, which may be used to formulate hypothesis (such as whether or not different groups have different learning rates - the exact focus of our study) [11].

2.3.1 Notable advantages of a computational approach

Whilst of course these models are limited, we are effectively testing a mathematically rigorous theories of cognition, the advantages of this approach cannot be understated. These models often succinct methods to:

1. ***Traverse neurological levels of abstraction***. Computational models allow us to better move between different levels of explanation. In neuroscience and psychology we examine phenomena from an extremely varied range of perspectives - from the synaptic level to observed behaviour - these model parameters may be interpreted at various levels; allowing for links between anatomical and social/behavioural phenomena [1].
2. ***Quantify the role of the environment***. These models allows us to explicitly capture environmental/social/external phenomena - better representing the state of reality as to be ignorant of our external environment is insufficient in scientific enquiry [1].
3. ***Allows us to better categorise the discrepancies between individuals***. Allowing us to design and implement very modular theories of cognition - thus learning distinct categories of pathologies that may be indispensable given our theoretical knowledge of the world [1].

3 Theoretically plausible models

Return to the observation that that firing of the midbrain dopamine neurons resembles a *reward prediction error* [3], we aim to capitalising on the rich reinforcement learning literature to design a comparable computational framework that offers rigorous scientific enquiry. The salient contribution of utilising a reinforcement learning framework is to explicitly model the dynamics dictating the *trail by trail* response to feedback, in contrast with traditional statistical techniques that emphasis modelling average behaviour [12]. This is indicative of the central issues of learning: how behaviour (or possibly neural activity) changes in response to feedback [3].

Data will often consist of a series of experimental choices and outcomes. In theory an arbitrary relationship can be used to describe the effect of outcomes on later choices, the task of the modeller is to design a parsimonious model that plausibly represents neurologically activity - attempting to capture some latent generative process [16]. This equates to modelling the expected state, action values $Q(a, s)$. It's worth noting that this approach allows one to quantify neuropsychological parameters that would in a traditional model but purely subjective - such as state value estimation, expectations, exploratory tendencies, risk aversion etc [3].

3.1 Theoretical vs empirical parameters

These models are often complex, verbose, systems however if we consider the model a module of statistical components one is able to search for a theoretically plausible, parsimonious algorithm. A combination of classical and Bayesian, though largely Bayesian, inference techniques are utilized to fit to the data [3].

These models are essentially a quantitative hypothesis about how the brain approaches a problem [3]. We wish to fit, compare and test an array of models with various levels of complexity and different covariate combination to arrive at a model that best articulates the underlying cognitive data generating process. Whilst models contain numerous free parameters, one salient distinction is the difference between:

1. ***Neuropsychological parameters***: infered by the data, these parameters (typically the learning rate α and exploratory dynamics β) describe the cognitive process.

2. ***Covariates***: as in standard statistical models, covariates offer the explanation of further variation in the data (often demographics of the participants).

Both parameter types are estimated by the data but are distinct in that *neuropsychological* parameters are purely a choice of the statistician designing the model whilst *covariates* are collected data that should be tested for explanatory value before inclusion - to adhere to the principle of parsimony.

3.2 Learning vs observation models

A further imperative distinction is the separation of computational theory into the:

1. ***Learning model***: describes the dynamics of the models internal variables - such as reward prediction error (RPE)[3].

2. ***Observation model***: describes the relationship between the internal, learning, model and the observed data - for example, how RPE drives choices or how prediction errors produce a neuralogical spike [3].

The observed model regresses the internal variables onto the observed data - regularly identical to the *link function* in general linear modeling [3]. The learning model is typically deterministic, however the observation model captures the stochasticity in the actual data, incorporating noise and assigning probabilities to the observations [3].

3.3 Parameter estimation

Suppose we have some model m, parameterized by a vector of free parameters θ_M [5]. In our case the model is the composite of the learning and observation models. The model M describes a probability distribution - or *likelihood* function $P(D|M, \theta_M)$ over possible data sets D [5]. Bayes' rule then allows us to derive the probability of the parameters having observed a data set D:

$$P(\theta_M|D, M) \propto P(D|M, \theta_M) \cdot P(\theta_M|M) \tag{1}$$

Formally, the *posterior* probability distribution over the free parameters, given the data, is proportional to the *likelihood* of the data, given the model and free parameters; and the *prior* probability distribution of the parameters [5].

Interestingly Bayes' theorem allows one to begin with a theory of some data generating process - that is a set of parameters that noisily produce data - and invert it into a problem by which data noisily reveal the parameters that produced it [3]. If we use an negate the influence of the prior, treating it as flat and uninformative, the most probable parameter estimates are those obtained by maximising the likelihood function $P(D|M, \theta_M)$ (that is the *maximum likelihood estimate* [3]. Let us denote this vector of estimates $\hat{\theta_M}$. Classical statistics is concerned with deriving these point estimates, often the maximum likelihood estimate, whilst this Bayesian paradigm is instread concerned with learning entire probability distributions over the parameter estimates [5].

3.4 Maximum likelihood estimation for RL

Consider a simple game in which an agent is tasked, at each trail t, with making a choice c_t between two options left L and right R. The agent stochastically receives a reward r_t after each action, where $r_t \begin{cases} 1 \\ 0 \end{cases}$. If we relate this dichotomous choice to Q-learning, the agent assigns an expected value to each machine $Q_t(L)$ and $Q_t(R)$ [3]. These values are initialized neutrally, say at 0, and then updated on each trail, which forms the *learning model*:

$$Q_{t+1}(c_t) = Q_t(c_t) + \alpha\delta_t \tag{2}$$

where $0 \leq \alpha \leq 1$ is a free *learning rate* parameter and $\delta = r_t - Q_t(c_t)$ is the RPE [16]. We then need to assume an *observation model* - to related the latent learning process to the observed data. It is natural, and frequently used, to assume that choices are made probabilistically according to a *softmax* distribution [3]:

$$P(c_t = L|Q_t(L), Q_t(R)) = \frac{exp\left\{\beta Q_t(L)\right\}}{\sum_{i=R,L} exp\left\{\beta Q_t(i)\right\}} \tag{3}$$

where β is a free parameter called the *inverse temperature parameter* that describes the agents willingness to *explore* vs *exploit* knowledge - the famous trade-off in search algorithms. β gives a weight to each choice value, such that agents with equivalent state-value estimates $Q_t(L)$ and $Q_t(R)$ but vastly different exploratory coefficients β may make greatly different choices, as this weighting describes how the agent samples actions [16]. It's worth noting this simple observation model is equivalent to a *logistic regression link function* where c_t is the response variable and $Q_t(L) - Q_t(R)$ is the independent variable and β is the regression weight coefficients [3]. It can also be shown that the above model is a special case of Kalman filters, a Bayesian smoothing technique that utilises the sample learning model but allowing for a dynamic learning rate α [3].

3.4.1 Likelihood function

Given the model described above, the data set D consists of an entire sequence of choices $c_{1...T}$ and the associated rewards $r_{1...T}$ - note that this describes a single agent. The likelihood function is the probability of the whole observed data set D and is given by product of their probabilities from equations 3 [3]:

$$\prod_t P(c_t = L|Q_t(L), Q_t(R)) \tag{4}$$

where Q_t estimates are determined by equation 2 given the observed rewards and choices. Equations 2 and 4 constitute the full likelihood, we can then estimates the free parameters $\theta_M = \langle \alpha, \beta \rangle$ by maximum likelihood [3].

3.4.2 Confidence intervals

Performing statistical hypothesis testing naturally requires confidence intervals around out parameter estimate $\hat{\theta_M}$. Intuitively, the reliability of the parameter estimate should be assessed proportionally to how probable alternative parameter choices are, that is the steepness of the gradient of the surface of the likelihood function [3]. The second derivative of the likelihood function with respect to the parameters - the *Hessian* - quantifies the steepness of slope in the likelihood surface [3]. The Hessian is a square matrix with a row and column for each parameter, with larger values indicative of a steeper slope. IF H is the Hessian of the negative log likelihood function at the maximum likelihood point $\hat{\theta}_M$, then it's inverse H^{-1} is the standard estimate for the covariance of the parameter estimates [3]. The variance of each parameter is then the diagonal of H^{-1}, thus the square root provide the standard error of each parameter. $\hat{\theta}_M + / - 1.96$ *standard errors* is then used to compute the 95% confidence intervals around parameter estimates.

3.4.3 Covariance between parameters

The diagonal elements of the inverse Hessian H^{-1} provide the covariance between parameter estimates, where larger values are symptomatic of multicollinearity and no unique optimum [3]. An additional complexity arises as when fitting Q-learning models because the reward $r_t is multiplied by both \alpha$ (when updating Q_t) and β (when computing the choice probability) before affecting the action a_t - making it very difficult to discern the effects of each parameter [3].Empirically, α and β estimates tend to be negatively correlated - inversely coupled - offsetting the effects of one another to reach a similarly likelihood.

3.5 Pragmatic implications of model fitting

Whilst the choice probability represent a regression model with a non-linear link function, standard open source maximum likelihood optimizers cannot be used because the Q_t values enter the model as a function of free parameters and thus are stochastic and do no enter the likelihood linearly and thus cannot be estimated by a general linear model [3].

Computing the likelihood given a parameter vector θ^i_M: given a data set - a sequence of *actions* a_t and *rewards* r_t - and a vector of parameters θ^i_M is it straightforward to iteratively cycle through the data, update Q_t and compute $P(c_t|Q_t, \theta^i_M)$; the product of which produces the likelihood function [3]. In reality, however, it is exceedingly likely that some choice probabilities $P(c_t|Q_t, \theta^i_M)$ are so small they exceed the floating point value of the computer performing the task. It is thus favourable to instead compute the numerically stable *log likelihood* $\prod_t P(c_t|Q_t, \theta^i_M) = \sum_t log\left(P(c_t|Q_t, \theta^i_M)\right)$ - a monotonic transformation with an equivalent maximum/minimum values [3]. In addition, the likelihood surface is invariant to any addition or subtraction of constant, thus under/overflow issue can further be mitigated by normalising the Q_t values, subtracting the mean from each value before computing the likelihood [3].

Searching the parameter space θ_M: One naive approach would be to discretize the parameter space of possible parameters to constitute θ vector and simply enumerate over the possible parameters, computing the log likelihood and selecting the parameter configuration that maximises the log likelihood. This is impractical for a multitude reasons: (1) as the number of free parameters grows, it becomes increasingly difficult - and often intractable - to compute all likelihood functions; (2) the granularity and boundaries of the search are predefined, leading too poor results or at best inefficient search [3]. In the case of non-linear models the tight coupling between variables further exaggerates these shortcomings [3].

Nonlinear optimization: A prudent alternative to grid search is to utilize a nonlinear function minimizer (thus requiring the negative log likelihood) - readily available in many open source software packages. Not only do these packages intelligently search the parameter space - efficiently sampling based on variations of hill climbing strategies - but also search continuously (without discretizing the space) which has the effect of increasing or decreasing granularity when required in order to reach better results [5]. It is important to remember that the search can often find a local minimum, thus is it prudent to stochastically or systematically initialize the parameter search many times and simply use the best run [3]. It also warrants noting that the Hessian or gradient vectors are often computed routinely by these non-linear optimizers, but in the event that they are omitted one can apply the chain rule to compute the matrices (for the purpose of confidence intervals) after searching the parameter space [3].

Bounded search: These nonlinear search algorithms often allow the developer to impose boundaries that, in theory, limit the space for efficiency. In the case of biologically plausible models, it is tempting to limit the search space to that that the theory allows (those that have semantic interpretations), in our case [3]:

- $0 \leq \alpha \leq 1$: learning rate. It is outside of the scope of this theoretical learning model to allow for large α values. Further large a large learning can lead to unstable estimates that grow rapidly and diverge [3].
- $0 \leq \beta \leq l$: where negative values have no interpretation large values l will likely lead to arithmetic overflow [3].

Boundaries should, however, be used sparingly and with caution for a number of reasons: (1) in

the case of a highly non-linear system the parameters are intertwined, thus constrains on one will effect the other; (2) optima is found outside of the theoretically plausible range may be indicative of a poor fit, noisy data or possibly the data revealing some truth outside of the current scientific consensus; and most saliently (3) most confidence interval estimates as well as model comparison techniques rely heavily on the inverse Hessian to compute the gradient of the likelihood surface - imposing boundaries will severely truncate or limit the likelihood surface and thus may impede the usability of these assessment criterion [5].

Bayesian regularization: It is of course, possible that the MLE reaches poor or uninterpretable estimates due to noise in the data, one flexible paradigm to regularize the parameters is to specify the model as a Bayesian system and utilize a prior $P(\theta_M|M)$ to constrain the possible parameter estimates [5]. This is, in fact, a generalization as specific boundaries can be show to be equivalent to some Bayesian prior - for example a hard $0 \leq \alpha \leq 1$ boundary is equivalent to simply imposing a uniform prior over the same domain [5]. If the Bayesian approach is adopted, one simply substitutes the MLE objective function (log likelihood) for the posterior from equation 1. Parameter estimates for a given participant may also be regularized by the broader sample of participants or groups in which they operate - a pooling variance technique known as Hierarchical mixture models, that is the focus of the following section.

3.6 Hierarchical models

The model discussed thus far is the choice behaviour of an individual participant, how then do we amend the model to account for multiple participants? One naive solution would be to average behaviour over all subjects - this, however, negates the importance of individuals differences, failing to address most interesting research questions. This approach treats all parameters as *fixed effects* that do not allow variation within subjects [3]. Instead, we aim to capture some variation *between* subjects [5]. Distinguishing between *within*-and-*between*-subject variability is of utmost importance in answering interesting questions; further, the failure to do so can result in overstated result significance [3].

Independent models: Another common approach would be to fit individual models and then use some summary statistics (such as average performance or parameter estimates) per model to test whether or not significant differences exist between subject (or possibly averages over groups) by utilising ANOVA, or one-or-two sided t-tests [3]. Treating each parameter estimate was a random variable - *random effects*. Whilst largely used and justifiable for many research questions, this approach fails to adequately capture the dependency between participants in a population, ignoring within-subject error bars (failing to smooth for irregular behaviour, noise and/or anomalies) [3].

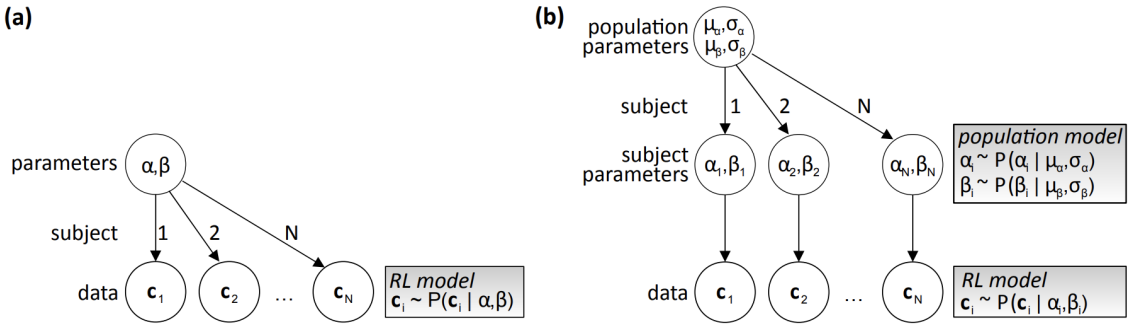

Figure 7: Capturing the inherent hierarchical structure of the data by imposing fixed vs random effects. (a) *Fixed effects*: parameter estimates are shared across subjects. (b) *Random effects* each subjects parameter estimates are drawn from a common population distribution - that becomes the regularizing prior [3]

.

Hierarchical structure over the population: It is possible to instead explicitly model how individual parameters vary across the population by imposing some distributional assumptions about the data generating process [6]. We can assume that, just as an individual is sampled from the population, an individual's parameters $\theta : \{\alpha, \beta\}$ are drawn from some population distribution over possible parameters [6]. For example, we may assume a given subjects parameters $\theta : \{\alpha, \beta\}$ are sampled from distinct Gaussian priors with some mean values μ_α, μ_β and variance values $\sigma_\alpha, \sigma_\beta$ - denoted as $P(\alpha \mid \mu_\alpha, \sigma_\alpha)$ and $P(\beta \mid \mu_\beta, \sigma_\beta)$ [3]. Further, parameter boundaries (such as the previously discussed $0 \leq \alpha \leq 1$ can be imposed by these probability distributions by limiting the supported range - in this case utilising a β distribution or normal distribution transformed through a logistic function [3]. These population distributions, of course, form regularising priors over the space of possible parameters [3].

The resulting two-level Hierarhical model now assumes data generating process whereby an individual's parameters α, β are sampled from a population and thereafter used to generate the data c_t by interacting with the environment (receiving stimuli and rewards r_t) [3]. The parameters of interest are usually the population level parameters $\mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta$ - as we are pooling variance to better estimate the populations behaviour - allowing us to answer questions about how the difference between population groups [3].

The probability of the observed choice behaviour of participant $i \in \{1 \dots N\}$ $\mathbf{c_i}$ (where $\mathbf{c_i}$ is a vector of choices over time for participant i) is then the probability given to them by the RL model $P(\mathbf{c_i}|\alpha_i, \beta_i)$ averaged over all possible hyper-parameter settings of the individual subject's parameters according to their population distribution [3]:

$$P(\mathbf{c_i}|\mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) = \int P(\alpha_i|\mu_\alpha, \sigma_\alpha)P(\beta_i|\mu_\beta, \sigma_\beta)P(\mathbf{c_i}|\alpha_i, \beta_i)d\alpha_i d\beta_i \tag{5}$$

Intuitively, equation 5 emphasises that from the perspective of performing inference on the population parameters, an observed individual α_i or β_i are nuisance variables to be averaged out [3]. The product over these individual distributions then gives us the probability of the entire observed data set (all N participants) [5]:

$$P(\mathbf{c_1} \dots \mathbf{c_N} | \mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta) = \prod_i P(\mathbf{c_i} | \mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta) \tag{6}$$

Bayes' rule is then used to recover the the population parameters given the entire dataset:

$$P(\mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta | \mathbf{c_1} \dots \mathbf{c_N}) \propto P(\mathbf{c_1} \dots \mathbf{c_N} | \mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta) P(\mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta) \tag{7}$$

Estimating population parameters in a hierarchical model: Utilising equation 7 we are now - in theory - able to estimate population parameters $\mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta$ using maximum likelihood (MLE) or maximum a prior (MAP) and estimate confidence intervals with the inverse Hessian - allowing one to compare between-group population estimates [7]. Implementable as follows:

1. Write a function that returns the log probability of choices over a population of participants given the population parameters $\mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta$ - equation 6.

2. Requiring computing equation 5 - providing the average over parameter values for a given subject.

3. This nonlinear system may be optimized through some standard non-linear optimisation algorithm.

As in almost all non trivial Bayesian applications, however, equations 5 is ***intractable*** and requires some approximate method [5]. A common remedy is to use sample techniques to estimate choice probabilities by the sample mean. One can draw k samples from distributions $P(\alpha_i | \mu_\alpha \sigma_\alpha)$ and $P(\beta_i | \mu_\beta \sigma_\beta)$; and use these samples to approximate the integral by averaging $P(\mathbf{c_i} | \mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta) \approx \frac{1}{k} \sum_{j=1}^{k} P(\mathbf{c_i} | \alpha_j, \beta_j)$ [5]. One pragmatic caveat is that off-the-shelf non-linear optimizers often require smooth parameter updating, which can become problematic with random sampling, thus it is recommended that that the same random seed is used for each subject iteration [3].

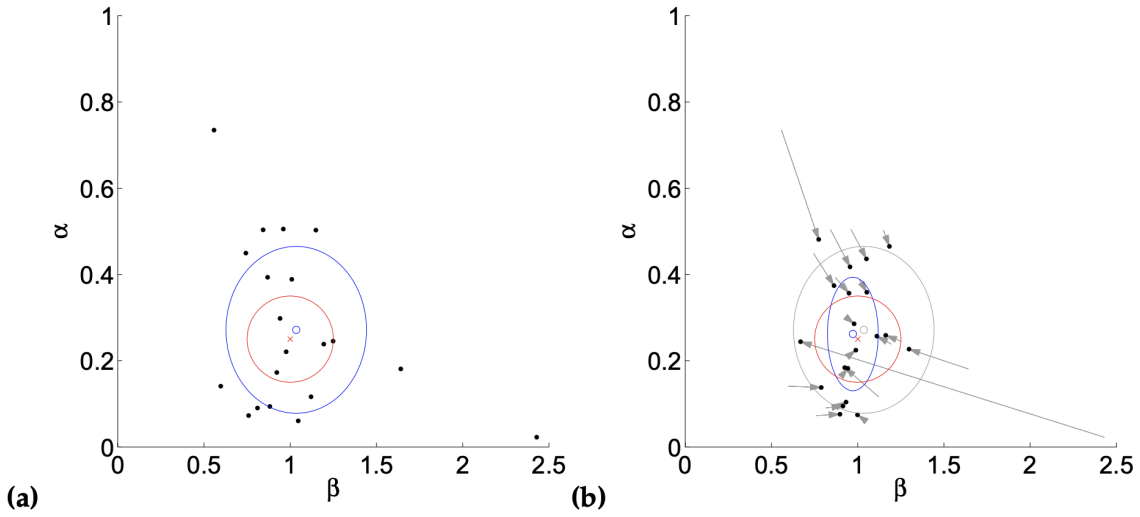

Figure 8: Simulated experiments detailing the benefits of a well specified prior distribution, where data are sampled from a bi-variate Gaussian and the true mean and standard deviation are depicted as red circles [**?**]. **(a)** Utilizes the individual/summary statistic approach whereby individual parameters are fit it each subject and then a Bi-variate Gaussian is fit to the population parameters by interpreting the individual subjects as samples; estimates are shown in blue [3]. Whilst the mean is well estimated, the estimate is unbiased, it appears to exhibit inflated variance [3]. **(b)** The individual estimates were now fit using MAP whereby the gray ellipse serves as the prior distribution, forcing the sample estimates towards true value, compressing the variance [3]. Imposing the prior is equivalent to fitting the hierarchical model whereby the prior regulates population assumptions.

Estimating population parameters via summary statistics: An alternative, and equally plausible, approach would be to simply estimate all the individual parameter $\theta_M^i : \{\alpha_i, \beta_i\}$ and thereafter make some distribution assumption about the data generating process - say assuming that individual parameters $P(\alpha_i|\mu_{alpha}, \sigma_\alpha)$ are drawn from normal distributions $\alpha_i \sim \mathcal{N}(\mu_{alpha}, \sigma_\alpha)$ (similarly for β_i) [5]. Thereafter it is straightforward to treat the individual parameter estimates as samples from the population distribution and fit Gaussian distributions to estimate the population parameters $\mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta$ [3]. Importantly, making these distributional assumptions allows one to use standard t-test and confidence intervals to test the significance of both between and within group variation - avoiding the need of computing the Hessian (second derivative of the likelihood function) for confidence intervals [5].

It is possible, that great noise can inflate the estimated population variance - as illustrated in figure 8 which warrants concern as this will greatly impact the usefulness of hypothesis testing [3].

Estimating individual parameter estimates in a hierarchical model: although in the above description the focus is estimating population parameter values and individual parameter estimates are treated as nuisance variables, some research questions requires individual estimates. Assuming population level parameters, it is possible to recover individual estimates [3]:

$$P(\alpha_i, \beta_i|\mathbf{c_i}, \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) \propto P(\mathbf{c_i}|\alpha_i, \beta_i)P(\alpha_i, \beta_i|\mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) \tag{8}$$

$P(\mathbf{c_i}|\alpha_i, \beta_i)$ is the likelihood of the individuals choice behaviour, and $P(\alpha_i, \beta_i|\mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta)$ serves as a prior, regularizing the estimate towards characteristics of the population [5]. The estimates are drawn towards that of the the group mean, thus the Bayesian equation details the balance between group level importance and individuality [5].

4 Incorporating additional data

Suppose, as in most experiments, we are able to capture some additional data about the subjects that may either:

1. Offer substantial explanatory power reducing variation in parameter estimation, or;
2. Allow additional detail hypothesis testing to measure the congruency with neuropsychological theory, asking whether or not observed covariates have an empirical relationship with choice behaviour $H_0 : \beta_j = 0$.

4.1 Explanatory power in the observation model

How then do we append these covariates to our model architecture? One would simply update the observational model to capture these parameter, defining the relationship between the data and the choice (response) [3]. The idea is that a common learning model may be observed by particular (observable) measurements given insight into the true underlying data generating process. One pragmatic illustration of this is to simply add Gaussian noise to the observed model, to better account for random fluctuations [3], however the idea is to add covariates to improve the model. Another common addition is reaction times (RT), for example:

$$s_t = \beta_0 + \beta_1 \delta + \beta_2 RT + \mathcal{N}(0, \sigma) \tag{9}$$

It has been observed in previous literature that if the primary focus is to test the relevance of additional covariates on learning or choice behaviour, one may wish to utilise a single global α (and other learning model parameters) to yield more robust, stable results [3] - which may be achieved by fitting a global fixed effects model to estimate some aggregate baseline learning rate parameter. This is, however, still very much an open question in the literature, however this may greatly deflate variability in the in individual parameter estimates (providing more accurate approximations) leading to more robust statistical diagnostics [3].

4.2 Explanatory power in the learning model

Of course it is straightforward to utilise alternative learning models, that may or may not include additional covariates (that then enter the model non-linearly). It is salient to consider where covariates ought to be amended, to achieve the optimal theoretical testing; for instance, adding an IQ parameter to the learning model may be appropriate as it might influence the underlying learning process [3]. As an illustration, one might assume that individual α_i values are sampled from a Gaussian where IQ effects the generation of the learning rate:

$$P(\alpha_i|\mu_\alpha, \sigma_\alpha, K_{IQ}, IQ_i) \sim \mathcal{N}(\mu_\alpha + K_{IQ}, \sigma_\alpha)$$

4.3 Alternative population models

The standard implementation described here assumes parameters are sampled from unimodal Gaussian distributions $P(\alpha|\mu_\alpha, \sigma_\alpha) \sim N(\mu_\alpha, \sigma_\alpha)$ however this too permits many variants and extensions [5]. One might imagine a situation where subjects cluster into copious groupings, this can be modelled by using a multimodal mixture model of the parameters [3], suppose there are two clusters:

$$\pi_1 N(\mu_{\alpha 1}, \sigma_{\alpha 1}) N(\mu_{\beta 1}, \sigma_{\beta 1}) + (1 - \pi_1) N(\mu_{\alpha 2}, \sigma_{\alpha 2}) N(\mu_{\beta 2}, \sigma_{\beta 2})$$

In this example μ's dictate the modal values and π the predominance of cluster 1.

4.4 Parametric nonstationarity

The models described thus far assume stationary (constant) model parameters throughout the experiment, which may be an unrealistic assumption for in many experiments. A high learning rate - whereby subjects update value estimates aggressively in an exploitatory fashion - promotes rapid acquisition but subsequent instability [3]. Similarly, a learning rate that adequately captures a subjects asymptotic insensitivity to feedback would predict an unrealistically slow initial acquisition [3]. It is natural to assume that rapid acquisition followed by asymptotic stability is desirable, ideally this would be fully captured by the converges of expected state values Q_t to approximations of the true (unknown) yield, however random sample fluctuations may produces this instability. This may be represented by an increasing softmax temperature, or decaying learning rate - with the learning rate decay capturing some value inertia [3]. It should be noted that this, again, alludes to the difficulty in heavily dependent parameters, an additional level of complexity whereby (in this case) entangles the change parameter relevance over time [5].

One approach to handling such nonstationarity is to model the dynamics of the non-stationary parameters by adding free parameters - expressing the system as a function of more elementary parameters [3]. Effectively adding granularity to the data generating process, allowing for variability in the learning rate. These, however, often adds superfluous complexity and further confound the parameter estimation [3].

If instead, we do not wish to specify a meta-level data generating process, another approach would be to allow for multiple learning/exploratory (α/β) parameters [3]. We cannot allow each trail independent parameters, as this would saturate the model (n parameters exceeding the number of samples - prohibiting unique parameter estimation) [5]. One option would be to assume the β_t parameters are variable but change linearly and deterministically, for instance:

$$\beta_t = \beta_{start} + \frac{t}{T}(\beta_{end} - \beta_{start})$$

Which would require learning a single additional parameter ($\beta_{end}, \beta_{start}$ as apposed to a single β) (and is a special case of the aforementioned strategy) [16]. One might wish to allow for some stochasticity, in which case the $beta_t$ parameters may be represented as some random process, for example: assuming β_t is captured by a Gaussian random walk [3]:

$$\beta_{t=1} = \beta_{start};\ \beta_{t+1} = \beta_t + \epsilon_t;\ \epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$$

Also containing 2 free parameters $\beta_{start}, \sigma_\epsilon$. One caveat to adding this stochasticity is the added complexity to model fitting: because the transition dynamics are probabilistic, behaviour in the expectation is estimated by averaging over many random trajectories to converge to a suitable

variance estimation - analogous to the aforementioned technique used over different subject-specific parameters [3].

Finally, a more deliberate approach is often taken, whereby experimental design is performed in an attempt to minimize nonstationarity. For example, in a multi-armed bandit problem, one may specify some stochastic defusion over reward probabilities - requiring the subjects to continue to learn the reward dynamics [3]. The approach taken in our experimental design - described below - employs this strategy by randomly changing the underlying learning rule, requiring frequent attention. Ideally, estimated learning rates should be asymptotically stable [3].

5 Model comparisons

Thus far model fitting has been discussed, however how should one select a candidate model in a plethora of possible variants, nestings and extensions? Further, to what extent does the data support different candidate models [3]? Many empirical scientific endeavours are inherently an abstract model selection process - as in the case of most neural studies - for the simple reason that the model defines the theory of interest. In our case, the optimal model corresponds to the best theory of the mechanisms behind human cognition, presupposing the question: *"What algorithm does the brain utilising to solve RL problems?"*, thus the model is tested by how well our theory fits the data.

When applied to Reinforcement learning, the most salient dichotomy is the distinction between model free learning (as performed in our standard Q learning model where candidates update state values directly); or model-based learning (whereby an agent evaluates actions indirectly by learning some more fundamental latent process and reasoning about said process [3].

Further, in general more free parameters will improve the fit of a model: a consequence of the classical curse of dimensionality that has been plaguing statisticians seen the dawn of empiricism [6]. Intuitively, added flexibility allows for greater overfitting, saturating the model.

Similar techniques from the aforementioned model fitting section are utilised to discern the optimal fit; a corollary of the fact that similar reasoning is used to derive the solutions; that is we aim to discover the best fit [3].

5.1 RL illustration

Policy and value models: When conducting model evaluation, we simply need to, again, compute data likelihoods under a model, optimize parameters and estimate Hessians [3]. One salient architectural choice when designing reinforcement learning systems is the discrepancy between *Policy and value models*. Formally known as the *representation* question: What is actually learned that guides behaviour [3]? *value based* models, such as Q learning, learn the value of actions; whilst *policy-based* algorithms learn the best course of actions directly to estimate the optimal choice strategy [16]. A simple instance of such a model, that updates the optimal policy directly, is illustrated here:

$$\pi_{t+1}(c_t) = \pi_t(c_t) + (r_t - \bar{r}) \tag{10}$$

As before, the choice behaviour is then captured by an observational model:

$$P(c_t = L|\pi_t(L), \pi_t(L)) = \frac{exp\left(\beta\pi_t(L)\right)}{exp\left(\beta\pi_t(R)\right) + exp\left(\beta\pi_t(L)\right)} \tag{11}$$

Where $\bar{r}$ is a comparison constant, often taken as the mean overall average reward and the model tracks a new free parameter π_t [3]. Equation 10 hypothesis a different model framework whereby the previous model estimated expected average reward Q for each choice and makes decisions based on relative value differences; π offers general "knobs" that control the actions taken [3]. This alternative model - defined by a unique set of constraints on the relationship between the feedback and subsequent choices - offers an alternative hypothesis of the data generating process [16].

It has been observed that choice values π of better than average options tends towards infinity $\pi- > \infty$, describing exclusive. choice of the richer option over time [3]. Q, however, tends towards the true unknown average reward value; allowing for less than complete preference for one option over others and thus implicitly capturing some quantity of uncertainty [16]. Undoubtedly these models will produce different choice predictions, however these particular predictions are only one aggregate feature of what are different trail-by-trail hypothesis about learning dynamics; as such the difference between models can be better and more robustly assessed by comparing their fit to raw data [3].

Choice autocorrelation:

Note that this alternative, policy based, model only has a single parameter β as apposed to the two Q-learning model parameters $\theta : \{, \beta\}$ supplying additional complexity in model comparison [3]. To illustrate the danger of saturating a model, consider the follow amendment to the to our standard Q-learning model:

$$P(c_t = L|Q_t(L), Q_t(L), L_{t-1}, R_{t-1}) = \frac{exp\left(\beta Q_t(L) + kL_{t-1}\right)}{exp\left(\beta Q_t(R) + kR_{t-1}\right) + exp\left(\beta Q_t(L) + kL_{t-1}\right)} \tag{12}$$

Equation 12 describes a model with binary indicator variables L_{t-1} and R_{t-1} that take values $L_{t-1} \in \{0, 1\}$ according to whether the previous trail's $t-1$ response was L or R - capturing an inertia term [3]. The motivation for this model is choice autocorrelation: whereby candidates have a tendency to either persevere with existing preferences or switch readily. Positive k values promote sticking, whilst negative values support alternation [3].

5.2 Classical techniques

How do we assess how well some set of model parameters fits the data? In some sense maximising the likelihood function achieves this, finding the parameter estimates M_1 that maximise the likelihood of the observed data $P(D|M_1, \hat{\theta}_{M_1})$ [**?**]. Although simple to compute, this approach inflates the measure of how well the model predicts the dataset [6].

Nested models: Returning to **??**, it is straightforward to see that this model is an extension to the previous discussed model. Thus dropping the binary indicator variables L_{t-1} and R_{t-1} and associated free parameter k yields the original model. Known as *nested* models, the former model M_1 is a special case of the later M_2 where $k = 0$, thus all parameter specifications available to M_1 is available to M_2 and thus M_2 is by necessity at least as well fitted to the data as M_1 [3]. Even if the data is generated by M_1, it is likely that noise is sampling the observations exhibit some bias towards a non-null k value [6]. In general, more complex models will *overfit* the noise in the data generating processes [6]. In the extreme case, the number of parameters equates or exceeds the number of data points, allowing for perfect (and completely non-general) interpolation of the data [6].

Crossvalidation: One common approach to address this concern is fit a model to a dataset ("training" set) and thereafter use another dataset (the "holdout", "testing or "validation" set) to

compute the likelihood of the testing set given the original (training) set parameters. If the model was greatly influenced by noise, it will fit the second data set poorly. That is, it will predict the second dataset well. Conversely, if the model adequately captures the true latent data generating process, it will predict the second data set sufficiently [6]. Since this approach is primarily concerned with predicting the held out data set, it allows one to compare models with different numbers of parameters - the holdout data set likelihood score is not inflated by the number of parameters (in fact, may be hindered by fitting noise) [6].

Whilst the prominent model section method of choice in many in many areas of neuroscience, it is not recommended that this approach is used in trail-by-trail analysis [3]. Given the temporal nature of the data, it is difficult to define a second, testing, dataset that is truly independent of the first [6]. Further, splitting the data temporally (training on earlier trails and testing on later trails) may lack the core assumption of being *identically distributed* - a consequence of non-stationary parameters in the data generating process [3].

Likelihood ratio test:

Consider, again, the case when a single data set is used to fit the maximum likelihood estimate. Although the estimate is inflated - often fitting noise from the given sample - statistical theory allows us to quantify the probability of this likelihood inflation [3]. We need to discern whether adding additional parameters to the model truly improves the fit, or if the improvement is a result of fitting noise when granted the flexibility of superfluous parameters [6]. If, and only if, we are comparing nested models a likelihood ratio test can be used to address this question. Under the null hypothesis that the data is generated by a simpler model M_1, rejecting the null hypothesis (in light of a low p-value) is interpreted as rejecting the simpler model with confidence. The likelihood ratio test statistic is calculated by fitting a complex model M_2 and simpler nested model M_1 to the same data set and thereafter computing:

$$d = 2 \cdot \Big[logP(D|M_2, \hat{\theta}_{M_2}) - logP(D|M_1, \hat{\theta}_{M_1}) \Big]$$

Since M_2 nests M_1, $d \geq 0$ by necessity [6]. The probability of a difference d arising from M_1 follows a *chi-square* distribution with degrees of freedom n (where n is the number of additional parameters in M_2):

$$d \sim \chi^2(n)$$

Therefore, the probability of the test: *the probability of a distance d or larger arising due to chance* is $1-\chi^2(d, n)$ [3]. Tested at some chosen level of significance, one can draw conclusions about the relevance of additional variables. Whilst powerful, and frequently used in regression analysis, the likelihood ratio test is both limited to nested models and primarily a frequentist methodology - both issues that may be addressed by Bayesian methods [6].

5.3 Theoretical Bayesian model comparison

Model evidence: When performing Bayesian inference, we are primarily concerned with computing the posterior distribution:

$$P(M|D) \propto P(D|M)P(M)$$

The probability of the data under the model $P(D|M)$ is known as model evidence [3]. It is salient to note that model evidence does not make any reference to any particular model parameters $\hat{\theta}_M$

[3]. It is for this reason that the score computed by a likelihood function $P(D|M, \hat{\theta}_M)$ is inflated by the number of free parameters: it takes as given parameters that fit the observed data [6]. Put succinctly, when asking how well a model predicts a dataset, it is a fallacy to retrospectively choose the parameters that would have best fit the data, after observing the data [3]. Overstating the models predictive capacity. When making model comparisons using model evidence $P(D|M)$ - agnostic of optimal parameters - negates overfitting. This quantity is computed by the (weighted) average of all possible parameter configurations for a given model, *prior to examining the data* $P(\theta_M|M)$ [3]. Formally:

$$P(D|M) = \int P(D|M, \theta_M)P(\theta_M|M)\, d\theta_M$$

Automatic Occam's razor: As a mathematical convenience, the posterior distribution favours simpler models. One might encode some regularising prior in $P(M)$, as if often done in Bayesian analysis [5]. External to this, inherent in it's formulation, the posterior computation has a preference towards simpler models by normalizing the model evidence $P(D|M)$ [3]. $P(D|M)$ is a probability distribution, thus must sum to 1 $\int P(D|M)dD = 1$ [6]. This means that more flexibly models (with more free parameters) assign lower probabilities to each $P(D|M)$ value (as they must sum to 1) and similarly simpler models assign greater probabilities to each $P(D|M)$ - effectively imposing a penalty on complexity [3].

Bayes factors: When comparing Bayesian models, the standard statistical quantity to quantify tow models relative fit is the ratio of their posterior probabilities (known as the Bayes factor):

$$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P(D|M_1)P(M_1)}{P(D|M_2)P(M_2)}$$

Conveniently, the Bayes rule denominator cancels out. The log of the Bayes factor is symmetric, positive and negative values favouring M_1 and M_2 respectively [3]. Although not identical to $p - values$, Bayes factors are often interpreted similarly. As a guiding heuristic: a Bayes factor of 20 (or log Bayes factor of ≈ 3) corresponds to a $20:1$ evidence in favour of M_1 which is analogous to a $p = 0.05$ [3]. A rich literature is available to convert Bayes factors to their familiar frequentist counterpart $p - values$ for ease of interpretation [3].

5.4 Practical Bayesian model comparison

Bayesian model comparison circumvents the disadvantages of utilising purely likelihood driven techniques, mitigating the risk of drawing ill-founded conclusion on inflated estimates [5]. These techniques, however, are accompanied with their own set of complications, namely:

1. ***Integrating the model evidence:*** As aforementioned, the model evidence is almost always intractable - requiring approximate estimates [5].

2. ***Prior specification:*** The assumed prior distribution plays a pivotal role is many Bayesian methods, and although flat/uninformative priors can by used in the absence of a defensible parameterisation, these neutral decision, too, bare consequences [5].

Priors: Prior distributions $P(\theta_M|M)$ act as a weighting kernel over the model evidence [3]. The prior, in the way, dictates the admissible range of the parameter space as well as the relatively likelihood of parameter configurations before seeing the data [5]. Put another way, the prior

regulates the parameter space by imposing (soft) constraints over the flexibility of θ_M. Moreover, because we are in the realm of probability density functions (requiring that values are normalised such that they sum to 1) ignoring the prior (as is done in the aforementioned techniques) is often both incorrect and mathematically unstable [3]. As shown below, BIC technique negates the need to specify a prior, however if one is able and will to specify some admissible prior distribution more favourite results can be achieved [3].

Sampling: As is often utilised in Bayesian statistics, the simplest method to approximate equation **??** is to average a sample [3]. This can be achieved by drawing candidate parameter estimates from the prior $\theta_M^i \sim P(\theta_M|M)$ and the compute the data likelihood $P(D|\theta_M, M)$ and averaging the results [3]. The avoids *optimisations*, but simple requires *evaluating* the likelihood at the sample points in the parameter space [5]. Although conceptually simple and easy to implement, this naive sampling technique is unadvisable for complex models: it is straightforward to see that as the number of free parameters grows, the curse of dimensionality ensures that the likelihood of sampling a suitable region of the parameter space diminishes exponentially [3].

Laplace approximation: One powerful technique is to approximate the function being integrated with a Gaussian - for which the integral can be computed analytically [5]. In this instance, we can approximate the likelihood surface as a Gaussian centred around the maximum a prior estimates $\hat{\theta}_M$ - notably, this is the same approximation utilised to motivate the reliance of the inverse Hessian H^{-1} to approximate error bars [3]. The Leplacian approximation applied to the model evidence is derived as:

$$log\left(P(D|M)\right) \approx log\left(P(D|M,\hat{\theta}_M)\right) + log\left(P(\hat{\theta}_M|M)\right) + \frac{n}{2}log\left(2\pi\right) - \frac{1}{2}log\left|H\right| \tag{13}$$

Where n is the number of free parameters and $|H|$ is the determinant of the Hessian - describing the covariance of the assumed Gaussian [5]. These quantities are straightforward to compute but do, however, require the specification of some prior over the parameter space (as computation is done with respect to the MAP estimate and not simply the MLE. Similarly, note that the Hessian is the Hessian of the posterior, and not merely the likelihood function.

One should note that equation 13 is the log posterior $log\left(P(D|M)\right) \approx log\left(P(D|M,\hat{\theta}_M)\right) + log\left(P(\hat{\theta}_M|M)\right)$ that is penalised by the last two factors $\frac{n}{2}log\left(2\pi\right) - \frac{1}{2}log\left|H\right|$ to account for the overparameterisation/inflated likelihood estimates [3].

BIC and related techniques: Bayesian Information Criterion (BIC) offers a simpler alternative, derived [3]:

$$log\left(P(D|M)\right) \approx log\left(P(D|M,\hat{\theta}_M)\right) - \frac{n}{2}log\ m \tag{14}$$

Where n is the number of free parameters and m is the number of datapoint (loosely indicative of the confidence of in the sample) [3]. Similar in that it takes a known computable quantity and adds a penalty for more complex models, it should be noted that BIC relies only on the likelihood $log\left(P(D|M,\hat{theta}_M)\right)$, not the full posterior, and thus is prior agnostic. Whilst convenient, neglecting the prior can result in far poorer results (given the critical importance of the prior as described above) [3].

It has been extensively shown that the Laplacian approximation more adequately estimates the model evidence and should be used if one is willing to declare and defend some prior over the parameter space [**?**]. These findings hold true in the space case of uniform, uninformative, priors

over a large range [3]. Parameters should only be penalised to the extend that they add explanatory power to the model: BIC blindly uses raw n and m values, irrespective of their actual fit, whilst the Laplacian approximation accounts for parameter uncertainty by the addition of the final term $-\frac{n}{2} log\,|H|$ [**?**].

One should also note that other penalised scores for model comparisons exist, most famous the Aikaike Information Criterion (AIC) $logP(D|M, \hat{\theta}_M) - n$ [3]. This, or related, quantity is irrelevant in our application as it cannot be used to compute approximate Bayes Factors, however may be used in adjacent applications [3].

5.4.1 Model comparison summary

It is perfectly plausible to compare likelihood functions directly (an intuitive solution as the likelihood directly measures the probability of the data given the parameter set) however, one would need to account for overfitting (fitting noise) when adding superfluous model complexity (free parameters) [3]. If models are *nested* the likelihood ratio test is a great method for comparing models that offers known statistical properties and thus an associated p-value test that is frequently used in the literature [3].

If models are not nested, approximate Bayes factors can be used as a basis of comparison. Given it's simplicity, BIC is widely used throughout the literature, however there is increasing evidence that - if one is able to specify some prior - Laplacian approximations may offer more reliable results [3]. Instead of relying on simple parameter/data counting (as done in BIC), the Laplacian approximation sufficiently accounts for variability of the parameter estimates (captured by the Hessian).

5.5 Model comparison of populations

How then do we extend these model comparison techniques to the aforementioned hierarchical structures so often used in Bayesian data analysis?

We need to begin by determining whether the model itself is a fixed or random effect [3]. If, as is often the case, we wish to make categorical claims at the mechanisms of the brain, it may be natural to assume no variability across subjects in the model *identity* (in opposed to in it's parameters). Following this logic, the model identity is then taken as fixed effect across subjects [3]. Of course, although the underlying data generating process may fixed across subjects, noise may enter the system at any number of stages resulting in unique fluctuations. Bayes' theorem then tells us:

$$P(M|c_1 \dots c_N) \propto P(c_1 \dots c_N|M)P(M) \tag{15}$$

Neglecting the hierarchical structure: One simple (and extreme) approach would then be to completely neglect any hierarchical prior and assume all individual subjects parameter values are sampled independently from some (known or unknown) prior distribution [3]. Assuming this independence, one can then decompose equation 15 across subject and perform inference separately (analogous to the summary statistics approach to parameter estimation):

$$log\,[P(c_1 \dots c_N|M)P(M)] = \sum_i logP(c_i|M) + logP(M) \tag{16}$$

This, naive but useful, approach computes the model evidence for the full dataset by aggregating the probability of the data given the model over each subject's fit (where this individual

subjects probability of the data given the model is captured by either the BIC or Laplacian approximation of the model evidence). Comparisons can then be drawn between models by using these aggregate values to compute Bayes factors [3]. Notably, if taking this approach one would likely report the number of independent subjects, and the proportion of individual subjects' models that are in agreement with the findings drawn by the population - to test the assumption of subject independence [3].

If the research question is primarily concerned with the population model, a more laborious - but perhaps robust - approach would be to integrate out thee population level parameters $\theta_{pop} =< \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta >$ [3]. Effectively computing the model evidence over all possible population level models [5], as detailed here:

$$P(c_1 \dots c_N|M) = \int P(c_1 \dots c_N|M, \theta_{pop}) P(\theta_{pop}|M) \;\; d\theta_{pop}$$

This quantity, of course, needs to be approximated by the aforementioned techniques (BIC, Laplacian Approximation, etc) [3]. Notably, the inner function $P(c_1 \dots c_N|M, \theta_{pop})$ again requires an integral approximation.

Lastly, one could consider some probability distribution over the model *identity*. That is, to assume *random effects* over the individual subject's model identity [3]. Whilst requiring an additional hierarchical structure to capture the variability across model identities, the implementation is largely unchanged.

It should be noted that a summary statistics approach to model selection - whereby researchers performing model section by averaging the Bayes Factors across individual subject models - is unfounded [3]. This is because of the nature of interpretation, this type of analysis would assume some variability model identity being sufficiently captured by that dispersion across subjects, an unjustified theoretically claim [3].

5.6 Salient caveats and notes

Here we leave some final remarks that may be important in both practical applications and in intuiting the aforementioned.

Why not assess models by counting the predictive accuracy? A nature approach may be to simply select the model that is able to best predict the (testing) data. This, however, is unjustified and is strongly advised against [3]. Firstly, this approach neglects the specification of a probabilistic observation model. Removing the reliance on statistical estimation forgoes the opportunity to make actual statistical claims (that is, inferring the results generalize to the population with sampling variability) [3]. Secondly, in our neuropsychological domain, we are frequently interested in model specification as a proxy for cognitive function. This too is abandoned if a pure predictive approach is taken [3]. Finally when employing this approach magnitude is completely omitted. Computing likelihood (or posterior) distributions allow one to quantify the degree to which the prediction deviates from the true response - as apposed to a binary indication - salient information about the variability of the system [3].

Is it possible to discern between-group parameter differences if parameters are correlated? As previously emphasized: although the learning rate parameter α and exploratory temperature parameter β may be independent in the underlying data generating process; they may be correlated when assessed empirically because they have similar expected effects on the observed data [3]. This may cause difficulty in interpreting the parameter estimates and testing quantities

across populations. One may wish to investigate what subset of parameters vary significantly across populations. This can be achieved by posing the question as a model selection hypothesis whereby various models are tested that share certain parameters (shared α vs shared β vs shared α and β) [3].

Assessing the model fit

Bayesian hierarchical inference lacks ubiquitous intuitive metrics to monitor model fit, however a number of model diagnostics are frequently reported.

Data likelihoods (often BIC-corrected) are regularly reported. The data log likelihood under pure chance can be easily computed [5]. These quantities allow us to compute the *pseudo-*r^2. A (theoretical) perfect prediction model producing a unit likelihood $P(D|\theta_M, M) = 1$. The *pseudo-*r^2 is computed as the fraction reduction in this log-likelihood of the model from the log-likelihood of the chance null hypothesis [5]. If R is the log-likelihood under chance (for example in a 100 trail binary choice task $100 \cdot log(0.5)$) and L is the log-likelihood under the fit model, then:

$$pseudo\text{-}r^2 = 1 - \frac{L}{R}$$

Because likelihood measures are usually aggregated across trails it can be more interpretable to average log likelihood per trial (i.e. $\frac{L}{T}$ for T trails). When dealing with choice data exponentiating this average log likelihood $exp\{L/T\}$ produces probability distributions that are readily interpreted relative to the random null model [5].

It is also straightforward to assess whether any model fits better than chance. Every model nests a 0-parameter empty model (which assumes all data is due to chance). A likelihood ratio test can be used to assess whether or not a model bests it's 0-parameter nesting [3]. A more rigorous, frequently used, test is test the full model against a nested alternative that contains only any parameters modeling mean response tendencies or biases - as is commonly done in regression analysis [5].

6 Non-linear optimization procedures

Finally, one must consider the search or optimization algorithm employed to fit parameter estimates. We are comfortable relying on the literature in selecting our optimization procedure.

This section will be complete after fitting the model, as we are not yet sure how it will be implemented. Pyro does, however, support a number of flexible nonlinear optimization strategies (like provided below).

1. A great explanation of EM is available here
2. Ridge regression
3. Pyro optimization options
4. Pyro custom inference

7 Ridge regression variable selection

Note: This section is, most likely, superfluous to our requirements and thus has not yet been completed and will probably be removed. Ridge/lasso shrinkage techniques are appealing as they maximise generalization, however as discussed above the nature of this data prohibits true independent train-test splits (a prerequisite for fitting hyperparameters utilised in shrinkage techniques). These requirements are circumvented by carefully pruning variables in the model free analysis, and then subsequently using model selection techniques (estimation of the Bayesian evidence and likelihood ratio testing) to find an sufficient model representation.

The above discusses allows for variable selection through a number of mechanisms, primarily: likelihood ratio tests for nested models and BIC or Laplacian approximations to estimate the model evidence.

These methods inherently require some step-wise procedure to add, remove and test variables sequentially. A suite of experiments may be designed and tested to select the optimal variable configuration, however this (and similar) approaches are not flawless:

- There is no guarantee that the subset selected from one step-wise procedure will emulate that achieved by another [6].
- If the model is saturated (there are more variables than observations $p > n$) backward variable elimination is infeasible [6].
- The minimum or maximum of a set of correlated F statistics is not itself a statistical quantity, such as an F statistic, that may be tested [6].
- Several different variable subsets may be equally valid - an outcome negated by step-wise procedures.
- There are several free hyper-parameters and decisions to made by the statistician: requiring a very manual, deliberate, architectural decisions where automatic solutions are often favoured.

A better solution may be to penalize uninformative variables in an automatic way by forcing superfluous parameter estimates towards zero: ridge regression. Not only does this approach allow for a single model fit, it also forces a smooth - more theoretically plausible and generalizable - likelihood surface.

7.1 Decision

Penalized LS requires hyperparameter tuning. Given the inability to split the data in train/test sets because of it's temporal nature, this is inappropriate. Alternative methods will be employed.

Ridge regression

End of chapter.

References

[1] Rick Adams, Quentin Huys, and Jonathan Roiser. Computational psychiatry: Towards a mathematically informed understanding of mental illness. *Journal of neurology, neurosurgery, and psychiatry*, 87, 07 2015.

[2] Tali M. Ball and Andrea N. Goldstein-Piekarski. Computational psychiatry: New perspectives on mental illness. *American Journal of Psychiatry*, 174(7):698–699, 2017. PMID: 28669207.

[3] Nathaniel Daw. Trial-by-trial data analysis using computational models. *Affect, Learning and Decision Making, Attention and Performance XXIII*, 23, 03 2011.

[4] Paul Gagniuc. *Markov Chains: From Theory to Implementation and Experimentation*. 05 2017.

[5] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.

[6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[7] Quentin Huys. *Computational Psychiatry*, pages 1–10. Springer New York, New York, NY, 2013.

[8] Quentin J. M. Huys, Marc Guitart-Masip, Raymond J. Dolan, and Peter Dayan. Decision-theoretic psychiatry. *Clinical Psychological Science*, 3(3):400–421, 2015.

[9] Quentin J.M. Huys, Michael Moutoussis, and Jonathan Williams. Are computational models of any use to psychiatry? *Neural Networks*, 24(6):544–551, 2011. Special Issue: Neurocomputational Models of Brain Disorders.

[10] David C. Knill and Alexandre Pouget. The bayesian brain: the role of uncertainty in neural coding and computation. *Trends in Neurosciences*, 27(12):712–719, 2004.

[11] George Konstantakopoulos. Insight across mental disorders: A multifaceted metacognitive phenomenon. *Psychiatrike = Psychiatriki*, 30 1:13–16, 2019.

[12] John P. O'Doherty, Alan N. Hampton, and Hackjin Kim. Model-based fmri and its application to reward learning and decision making. *Annals of the New York Academy of Sciences*, 1104, 2007.

[13] Thomas Parr, Geraint Rees, and Karl J. Friston. Computational neuropsychology and bayesian inference. *Frontiers in Human Neuroscience*, 12:61, 2018.

[14] Wolfram Schultz, Peter Dayan, and P. Read Montague. A neural substrate of prediction and reward. *Science*, 275:1593 – 1599, 1997.

[15] David Silver. Lectures on reinforcement learning. URL: https://www.davidsilver.uk/teaching/, 2015.

[16] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[17] Csaba Szepesvari. *Algorithms for Reinforcement Learning.* Morgan and Claypool Publishers, 2010.