The Mathematics of Cognition: Inference, Reward, and Behaviour

Dalton A R Sakthivadivel

30th July 2020

Contents

1 Introduction

Here are my lecture notes for Neuromatch Academy 2020, Week Two. It consists of reviews of some of what we covered and new presentation of the content. I've tried to structure these notes to emphasise portions as they relate to neuroscience or psychology. I've combined this with some necessary mathematics, while also trying to emphasise the connections between topics to place them in some coherent, overarching mental framework. As such, these notes are aimed primarily at neuroscientists, psychologists, cognitive scientists—not physicists, mathematicians, or statisticians—but there will be a bit of maths here and there.

I hope you enjoy learning about these things, and I hope you felt like you got something useful from NMA, even if it was simply exciting discussion, or a new appreciation for statistics or for neuroscience.

With that, let's begin by looking at inference.

2 Bayes' Theorem—A Probabilistic Interpretation

2.1 What is Inference?

Inference is, in the logical or epistemological sense, a conclusion reached on the basis of evidence and reasoning. Given some evidence, we may infer something about what we are looking at—for example, Sherlock Holmes solved crimes by induction, a method of reasoning where the truth of the conclusion of an inductive argument is probable, based upon the evidence given.

In some cases, when we are viewing a system, we are only able to observe something—e.g. make a measurement, or experience some sensory perception. In the absence of noise or other latent variables which might affect this, this would be perfectly fine, as this assumes a one-to-one mapping between our observation and the system that produced that observation. However, given some noise or some variable that affects the system output, we can no longer be sure that our observation reflects the system faithfully. How do we know anything about the system then? Is our reality actually our reality?

Luckily, we can utilise a number of inference techniques to say with exact certainty what we do indeed know about the underlying system giving rise to our observations. Inference allows us to 'lift the veil' in this very specific sense.

2.2 What is a Prior?

In the broadest sense, a prior is some previous sense of the system we are observing. It is intuition, based on experience. Formally, a prior probability distribution is the probability distribution that would express one's beliefs about this quantity before some evidence is taken into account. If we have a model about how a system behaves, or in other words, a long term inference, it is the probability that the inferred conclusion is true on its own.

2.3 What is a Posterior?

A posterior indicates we are evaluating our model in light of some evidence, in the form of $P(I \mid O)$. This is the probability of our conclusion given some evidence—a data point, or an observation, perhaps. Or, in the case of iteration to perfect an inference, it could be a new observation, arriving after we have adjusted our model to increase our posterior.

2.4 Bayes' Theorem

Bayes' theorem finds our posterior, or our inference given an observation. In order to find this, it multiplies the likelihood function (which you should be familiar with by now—given an inference we want to fit to data, we go in reverse, fitting our data to the inference to determine whether our data is likely to be true if our model is absolutely true) with our prior model, and puts this over the probability for our observation. This is indicated by

$$P(I \mid O) = \frac{P(O \mid I) \cdot P(I)}{P(O)}.$$

Where do these components come from? The likelihood gets multiplied by the prior as a result of a simple mathematical rule. The following axiom is true, probabilistically:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)},$$

or, that the conditional probability of any event A given B is equal to the probability of A and B over the probability of B.

Keeping the notation for our inference making, model building framework, and the associated meanings, it states that the probability of our inference being true given our observation is the probability of our observation and our inference both being correct, normalised to the probability of our observation being correct (so as to be robust to false conclusions).

We may apply the Product Rule here to determine the following:

$$P(A \cap B) = P(B \mid A) \cdot P(A)$$

and so we arrive at Bayes' theorem by substituting the above expression into our conditional probability.

Here, we are saying that in order to evaluate how good our inference is if we have an observation, we need to consider the probability we would see our observation if our inference were indeed correct, the probability that our inference is correct, and the probability that our observation is not an outlier. It follows from a couple of simple lemmas about the probability of events, and yet, has a very powerful sole interpretation.

We may examine this further if we take the following: consider the set of events (possible inferences, or explanations, for our measurement) we are evaluating, $I_1, I_2, I_3, \ldots I_n$. We say that this set of events partitions our sample space S. If this is the case, then the following is necessarily true:

$$P(O) = P(O \mid I_1) \cdot P(I_1) + \ldots + P(O \mid I_n) \cdot P(I_n).$$

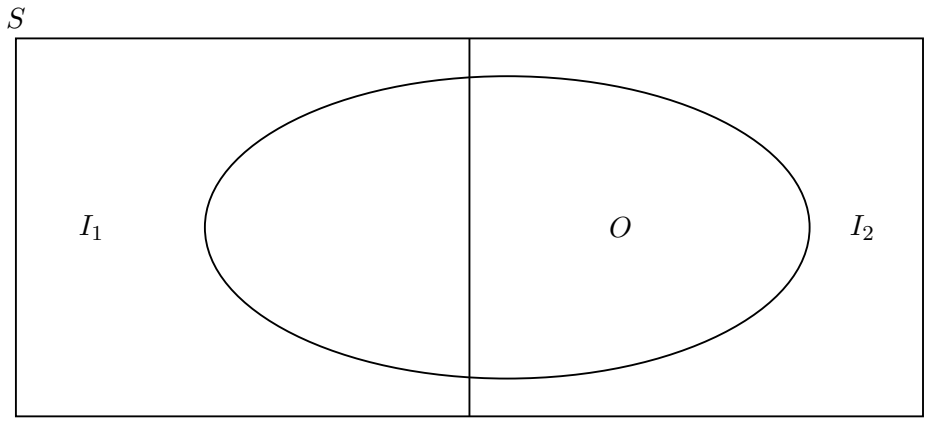

Figure 1: **The Partition Theorem.** A geometric proof of the partition theorem shows how an observed set of values sampled from a sample space S decomposes over inferences.

This is the law of total probability, or the partition theorem, and the intuition behind it is simple: it states that if one sought to find $P(O)$ as it depended on a number of different other variables, or partitions, they could look at a partition of the sample space S in which these variables depend

on each other, and add the amount of probability of O that falls in each partition.

From here we can rearrange Bayes' theorem in such a way as to consider competing inferences at their respective partitions, thereby evaluating multiple possible inferences:

$$P(I_n \mid O) = \frac{P(O \mid I_n) \cdot P(I_n)}{\sum_i P(O \mid I_i) \cdot P(I_i)}.$$

Once again, the utility of Bayes' theorem is clear. It allows us not only to make, but to find optimal, inferences.

2.5 Tutorial Recap—Week Two Day One

On Monday, we began the week by looking at Bayesian Statistics. Bayes' theorem allows us to make an 'inference' about something, or an estimate about the true value of some hidden variable of something, based on some model of the environment. Let's lay some groundwork for how the brain processes stimuli, and then use it to understand what is going on in this tutorial. After that, we'll go through some of the important functions to gain some insight into how we actually implement this.

Some notes on the tutorial content for the day are in this section, as a companion to Week Two Day One.

If you are unclear about the point of Bayesian inference for cognition, it may be wise to skip to section 3.1 where this is elaborated on.

Here we want to develop a Bayesian model for localising sounds based on audio and visual cues. This model will combine prior information about where sounds generally originate with sensory information about the likelihood that a specific sound came from a particular location. We are using a Gaussian distribution for our prior because it describes our problem well—we have a peak where something is most likely, and a spread of other likely values that compete for our certainty estimate.

In the first function we use a general form of the Gaussian function, and so we normalise to restore it to a probability density (which has the necessary condition of integrating or summing to one—mathematically, we can't have a density that encodes probabilities of events exceeding one). We take the sum of all the points and we scale our density by that, so that each point in the density is some proportion of one and thus each proportion adds to one. An important note here is that one would typically normalise by area, not total. See, for example, this: *How to normalise a histogram in MATLAB?* This is related to the fact that, despite being described the

same way, an integral is fundamentally not the same thing as a sum, and so an area is not the same as a total. This fact is given by Fatou's Lemma, among others.

In our next function, exercise 2A, we multiply our various distributions to find our posterior. We normalise again.

Next, we work with our posterior a little bit.

Our bonus looks at a bimodal prior, or two peaks—two locations where we think our sound would likely be coming from. Looking at the graph, we see our blue curve—the visual data—is closer to one peak of the prior, corresponding to one spot we expect things to come from. This shifts our posterior towards that peak (at $\mu = 3$). The graph below it shows the behaviour of our posterior for various visual peaks. When the visual peak is closest to either prior peak, our posterior says we are most certain of that location. When the visual peak is right in the middle, at 0, then we're not sure of either one.

Moving on to tutorial number two, we're doing a slightly more complicated inference. We're no longer looking at what the brain is doing, as in tutorial one—we're harvesting priors from the brain and trying to infer what the subject is thinking using Bayes' theorem. It's a different end goal now. The text in section one covers this in greater detail.

Because our subjects learned there are two options—sound comes from the poppet or from somewhere else—with a given likelihood—75% and 25%, respectively—we need to build our prior accordingly. We have a prior with a weighting p given to our prior distribution for common sources, and the remaining $1 - p$ is given to independent sources. We build this mixed prior in such a way that one distribution encodes both options. Could we have used a biomodal distribution for this? As long as were sufficiently careful with the weightings as likelihoods, presumably, we could. A mixed prior is much more useful here though, as one of the things we are trying to infer is the parameter set that our subjects learn, or the weighting they ascribe things.

The interactive demo allows us to look more at how this mixed prior affects our posterior. The first observation is that σ_{common} has a much larger affect on our prior than $\sigma_{independent}$. This is because our weighting for common sources is much higher. Our posterior doesn't move much—that's because our $\sigma_{auditory}$ is very low. We are very certain of what we are hearing, and it could be that this surmounts even scenarios where we have high uncertainty in our prior about where the sound might be coming from. Playing with this interactive demo will provide very good insight into how a prior affects a posterior, and thus, how learning affects estimation.

Tutorial number three is a dense one, so I'll go through it a bit more carefully.

We begin by presenting a stimulus to the participants—this is x, the position of which is known to the experimenter, but unknown to the subject and their brain. The brain has some encoding of x, $\tilde{x}$, which is given by the true x. Using the likelihood of a Bayesian model, the brain finds what the encoding of x is likely to be based on the given stimulus—$P(\tilde{x} \mid x)$. The brain has a prior for the location of x in the form of $P(x)$, and the brain combines all of this to form an estimate of the true location of x in light of some known encoding, which is our posterior, $P(x \mid \tilde{x})$. A response, $\hat{x}$, occurs based on the posterior estimate.

Here, subjects need to find the location of a sound given the same prior information as tutorial number two.

We will first pretend to be a brain and craft our likelihood. We want to find the likelihood that our encoding is correct, and to do so we plot each encoded x, $\tilde{x}$, with its given x.

In the plot in exercise one, we see that there is a more or less $\tilde{x} = x$ sort of relationship between the two—this is good, because it means we're encoding mostly correctly. There is also a very small variance in this encoding given an x, which means our encoding is certain.

We will make our prior next, using the same procedure as tutorial two. We're going to tile it so we have this prior for each of the likelihoods we calculated previously. We are storing it in an array to make the associated multiplication easy—we just multiply an index of one array with the other, for every shared index.

If we look at the following plot, we will see the prior does not depend on the encoding, and is centred at 0 for x's true value. This means our prior is independent of our likelihood and places x as coming from where x actually is—this is good, because it means we assume we hear things from where they are coming from, rather than from the place next to the source or above it. If our clock is chiming the hour, and at the same time, a person is talking to us, it wouldn't be very good to hear this person go 'bong' and the clock ask us how our day went. Good priors are how we cope with a confusing world.

We do this multiplication to find our posterior and plot it, in the exercise below. This gives us our plot for where we estimate x to be if we have some encoding of x, $\tilde{x}$. It is clear that our prior has a large affect on our estimate—we are most certain when x is closest to our prior of where x is—zero. As the true x gets further from our prior we are less and less certain—if we think of this as confusion about what we expect and what

the data is saying, this should make sense.

The break, where the centre region of high certainty goes more vertical, indicates that when our hypothesis corresponds to where x actually is (zero offset, as measured by the x axis), our estimate depends less on our encoding. This is also very good—it means that our estimate is robust to bad encodings when our prior is strong enough. Even given a false encoding like -1, we are still correct about x being 0 (on top of the true x).

So now we have made a posterior for where x should be given many different possible x values. This doesn't necessarily estimate x for us though—we've only collected a bunch of information about different x and $\tilde{x}$ pairs. We will now make an estimate of the position, which we will observe by proxy, looking at the subjects response $\hat{x}$. To do so, we will look at each encoding and find the mean—the estimate with the most certainty. The point associated with that estimate will go into our plot. We will proceed by placing a one in that index and a zero everywhere else.

We see this takes exactly the shape of the centre of our posterior plot—this is to be expected, because in taking the estimate of x, $\hat{x}$, we are taking the point of greatest certainty associated with the posterior plotted at x. In other words, we are taking the centre of the line.

The remainder of the tutorial is where things get a bit strange.

Remember, we're not being brains now, only pretending—in reality the brain would make a single estimate given a single observation, not a number of estimates for every possible pair of encodings and every possible x. This is simply a model of what's going on, and there are some artificial steps that need to be taken, like generating a lot of input-output pairs so as to find the patterns in the data.

The next step, in exercise five, is simply showing what our is encoding when we use a single experimental stimulus with position 2.5. The respective encoding is fairly constant at $\tilde{x} = 2.5$, so we're happy with our results—it means in our experiment where we use 2.5, our subject's brains are likely to encode the right value, no matter what other hypothetical stimuli could have been shown. If this seems backwards, don't worry about it too much. What this means in more explicit terms is we want the likelihood that of $\tilde{x}$ being compatible with the given x. In $P(\tilde{x} \mid x = 2.5)$, x is the true x set by the experimenter, and the horizontal axis has the one hypothesised to be true by the subject so that it is compatible with $\tilde{x}$. We only really need a single argument marginal distribution, but to make computation more easily visualised, we replicate it with a set of hypothetical x's (hypothesised by the brain), which are independent from the true x.

This will be used as our input array. In so doing, out of all the possible

pairs we produced earlier, we are keeping only the encodings and estimates that correspond to our actually given stimulus.

In our marginalisation step we are looking at finding $\hat{x}$ from our density characterised by the measurement of our estimate (our response, $\hat{x}$) given our encoding. We want to marginalise this latter variable out to get $\hat{x}$ from our real x. This is for us, the experimenter, to help determine what is going on inside the brain—when the brain gets a stimulus of 2.5, what does the response that reflects the estimate of the stimulus look like?

Our final step, in section seven, is to fit this model that we have created to an actual subject who has been presented a stimulus 2.5 away (an independent source). In doing so, we'll be able to fit our model architecture to their output, and assuming our model describes the person accurately, we'll be able to find the participant's 'parameters' to determine how they have made decisions. This is for the experimenter.

We're performing *model inversion*, so as to test our model. Recall our generative model for maximum likelihood estimate, which we resurrected as a measure of model 'correctness' later when we looked at linear regression concepts (e.g. GLMs). You'll note the docstring for my_Bayes_model_mse() is 'function fits the Bayesian model from Tutorial 4.' What, precisely, are we fitting, and how is it driven by a likelihood estimate? Earlier in the tutorial we mixed our priors according to some weighting, so as to exhibit a preference for one prior or the other. We use the usual generative model process—generate a bunch of data for each parameter and see which matches our true data, because that implies we have found the true parameter set, if we assume a unique mapping between features or parameters characterising a data set and the data it characterises.

Tutorial number four introduces a cost function for our estimate—a penalty is now associated with a low certainty inference. This will allow us to perfect our inference, if we continue to make it so as to minimise that cost function.

3 Bayes' Theorem—A Cognitive Perspective

3.1 Model Building and Inference

Why does the brain use Bayes' theorem, and how does it do so? The key is model building for environmental inference. The brain is well aware that it observes the world around it in the presence of noise, in addition to storing information as an encoding, rather than the truth of a thing. To get around this, it builds a prior model based on experience about what

inferences have been correct or not correct. It uses this model to infer the cause of some observation, in order to connect the observation to the source of the observation.

Take a simple example related to what we examined in the tutorial. A person hears a noise in the woods. They can't be sure where the noise came from—the sound waves have bounced all about, diffracting around trees and being dampened by the heavy thicket. Luckily, the brain has made many inferences about sound travel over the years, and has built a prior model for how woodland scrub affects sound travel. The brain will now make an inference about the location from where the sound was coming from, and then evaluate this inference by visual confirmation—if I see a the source of the sound, I can say with certainty that it has come from the location I inferred it from, thus my posterior is high. If I don't see it, I can revise my model and the inference that comes from it, in order to re-infer, and check again.

Here, the inference comes from experience, an iterative, self corrective form of Bayes' theorem. We build a model which yields an inference, so they are functionally synonymous at any given point; however, they represent different things in the long term. Bayes' theorem as a statistical method is often an inference in a single point in time, telling me, simply, what is the probability that a measurement has come from some underlying data. In this case, there is a reward for building a good model, as we learn from inferences both good and poor. Suppose the sound was a large, relatively cross, brown bear—a good model can often mean the difference between life or death. In later sections, we will examine reward.

3.2 Bayes' Theorem and Theoretical Psychiatry

There is an increasingly large, though still relatively new, body of work investigating the theoretical underpinnings of psychiatric disorder using this Bayesian model of cognition. Various aspects of things like delusions, psychosis, schizophrenia, and more can be connected to misprocessing of information or prediction error in the sense of Bayesian model building. I highly recommend the work of Paul Fletcher and Phil Corlett as it relates to this subject. Three good articles on this are included below.

Katthagen et al, 2018. Modeling subjective relevance in schizophrenia and its relation to aberrant salience. PLOS Computational Biology, 14(8).

This study shows one element of misdirected or malfunctioning salience in schizophrenia is due to poor estimation of importance for some data,

based on poorly updated priors.

'We found that subjects use Bayesian precision to estimate stimulus relevance in order to integrate multidimensional information and adapt more to the subjectively relevant stimuli... To conclude, our findings demonstrate how individual beliefs about relevance can be inferred from computational models. Furthermore, we suggest that aberrant salience observed in patients with schizophrenia reflects an idiosyncratic bias in states of high subjective uncertainty.'

Fletcher and Frith, 2009. Perceiving is believing: a Bayesian approach to explaining the positive symptoms of schizophrenia. Nature Reviews Neuroscience, 10(1), 48–58.

This model states that poorly functioning prediction-error needs some other intervention to adjust properly. This proposes a two factor model of schizophrenia, wherein false perceptions modulate further false beliefs, by model dysfunction.

'Recent advances in computational neuroscience have led us to consider the unusual perceptual experiences of patients and their sometimes bizarre beliefs as part of the same core abnormality—a disturbance in error-dependent updating of inferences and beliefs about the world. We suggest that it is possible to understand these symptoms in terms of a disturbed hierarchical Bayesian framework, without recourse to separate considerations of experience and belief.'

Fleming et al, 2020. Drugs That Induce Psychotic Symptoms Acutely Impair Mediated Learning in Rats. Biological Psychiatry, 87(9).

In this brief report, Leah Fleming, Phil Corlett, and their team inventively show that a relationship exists between learning and psychosis by administering drugs which create symptoms of psychosis and observing a dysfunction in prediction based learning.

3.3 The Free Energy Principle for Action and Inference

One of the most popular theories of Bayesian Cognition today is Karl Friston's Free Energy Principle (FEP). I've endeavoured to explain it here.

One thing to note immediately is that this is a highly complex topic, even for experts, even at an intuitive level. As Peter Freed of Columbia's department of psychiatry said, in his 2010 Research Digest for Neuropsychoanalysis:

> 'At Columbia's psychiatry department, I recently led a journal club for 15 PET and fMRI researchers, PhDs and MDs all, with well over $10 million in NIH grants between us, and we tried to understand Friston's 2010 Nature Reviews Neuroscience paper — for an hour and a half. There was a lot of mathematical knowledge in the room: three statisticians, two physicists, a physical chemist, a nuclear physicist, and a large group of neuroimagers — but apparently we didn't have what it took. I met with a Princeton physicist, a Stanford neurophysiologist, a Cold Spring Harbor neurobiologist to discuss the paper. Again blanks, one and all: too many equations, too many assumptions, too many moving parts, too global a theory, no opportunity for questions—and so people gave up.'

Let this not scare us off, lest we also make such a mistake as giving up. Instead, keep in mind that it's a difficult topic, and often poorly explained. It's perfectly fine to not be completely sure of the 'why' or the 'how'.

We should begin with the matter of sensory inference. Inference comes from models—a prior for what is a likely inference, or some experience about how a system behaves in order to connect evidence with the sensory system it interacts with. In other words, rarely does the brain simply take a guess. Unlike what could be called 'standard' Bayesian inference, which makes an inference and gives it a certainty, the inference we make is guided by a perfected prior over time, to maximise our posterior in a given instant. This is guided by prediction error, which we will examine later.

FEP posits that our brain can be modelled as having an internal state dependent on sensory information. This is trivial—sensory information determines what our brain is doing. Are the neurones in our visual system firing in response to some stimulus? Is our Default Mode Network (resting state) activation higher than it was previously, now that there's no stimulus? And so on. Additionally, from a thermodynamical perspective, information is a physical construct, which can exert change on a system it is entering. This is how entropy is measurable, or how disorder propagates in a closed system. Since a state is merely a descriptor of the system, in the brain, states depend on information.

Building on this foundation, FEP says that when information is surprising, this is a bad thing. This is indeed surprise in the entropic, thermodynamical sense. When information exerts a change on our internal state, the more surprising this information is, the more change occurs. Surprising information causes disorder—this is because surprise is intimately linked

with randomness and disorder. Disordered brain states mean death. We want to therefore place a bound on disorder in the brain, which does two things—makes information less surprising, implying an improved model; as well as placing a restriction on the number of states the brain can be in for optimal performance. FEP is indeed related to a control theory approach, which is something we've seen before—FEP says that surprise and therefore states must be properly regulated.

Rather than using surprise, which is a hidden variable, we use free energy, which is a function of sensory states and internal states. Minimising free energy allows us to bound surprise and thus maintain some order in our brains. Free energy is at a minimum when our model, encoded in our internal states, is correct (or corrected). There are two ways of doing so—learning (revising our model in response to our environment) or action (changing our environment to agree with our model).

This is rather a high level overview, with the interest of not becoming overwhelming. We've inadvertently touched on reward and prediction error, learning, decision making, action, information theory, and the thermodynamical nature of cognition. As one can tell, FEP is very powerful as a unifying explanation for many phenomena. Unfortunately, there are many critiques of it, and it has not much to say in response. In my opinion it has a central flaw, which is that it inherits the flaws of the theories it seeks to explain. Issues such as the Dark Room Problem and metaphysical dualism, as well as questions about the assumptions that have gone into the mathematics behind it, are difficult for FEP to surmount, despite ongoing work to perfect it. It is, nonetheless, extremely exciting, as it represents a uniquely large advancement in the theory behind theories of cognition, learning, and more.

Some sources referred to in this section are the following:

For the maths behind FEP, see this article, which is somewhat friendly to non-maths people and fairly well explained: *Bogacz, 2017. A tutorial on the free-energy framework for modelling perception and learning. Journal of Mathematical Psychology, 76(B), 198-211.*

For a physics based, but non-mathematical, overview of FEP, see: *Friston, 2010. The free-energy principle: a unified brain theory? Nature Reviews Neuroscience, 11(2), 127–138.*

For an article with more focus on the psychological implications of FEP, see: *Freed, 2010. Research Digest. Neuropsychoanalysis, 12(1), 103-106.*

Wired wrote an article exploring FEP in 2018, located here: *The Genius Neuroscientist Who Might Hold the Key to True AI.*

4 Markov Processes

4.1 Inference Revisited and Markovian Priors

Previously, we took pains to note that there is no necessary requirement of 'correctness' placed on Bayes' theorem. Bayesian inference usually makes sufficiently good single inferences given a sufficiently good prior. However, this prior could be a single idea, not necessarily perfected over time; similarly, the inference could be quite uncertain.

There is more than one way to infer the underlying dynamics of a system, and luckily, for some, we can utilise notions of temporality more explicitly.

A Markov process is a system that hides a true state X behind an output Y, in such a way that the current time-step t depends *only* on the previous time-step $t-1$. It is thus a 'memory-less' system. We performance inference on these systems using Markov chains or Hidden Markov Models. Often, Markovian inference is in the form of finding the truth—what is X if I know Y? It could also be in the form of building knowledge to predict something—if I know Y_t, and can estimate X_t, what can I say of X_{t+1}?

As was stated, we use Hidden Markov Models (HMMs) to perform inference on such systems. In a HMM, the following rule is true, such that this is the only useful way to work with these systems:

$$\hat{x}_t = P(X_t = x_t \mid X_{t-1} = x_{t-1}),$$

or, that our estimate of x at time t can only be from the prior probability of X_t being some value given the previous value of X_{t-1}, x_{t-1}. This is a formalism of mine, rather than an algorithm—this prior remains to be found.

We can do so using a recurrence relation. In the first tutorial from day four, this is defined as follows:

$$P(x_t \mid y_{1:t}) = Z^{-1} \cdot P(y_t \mid x_t) \cdot \sum^{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_{1:t-1}).$$

What this means is—given all the measurements, or observations of a system's output, we have made thus far, we can predict what the state at time t will be. We build an 'intuition' for how the system behaves over time,

in an effort to connect measurements to states and thus states to successive states.

4.2 As Compared to Bayes' Theorem

There are clear similarities to Bayesian inference, despite key differences. Both make inferences given a prior. Both attach a quantified, rigorous idea of uncertainty to their estimates of some latent variable. In some cases, Bayesian inference builds a prior over time. The key differences are primarily mathematical. A Markov process is necessarily only defined relative to the time-step directly previous—nothing more, nothing less. An iterative Markovian model can overcome this, but this has implications for how Markov chains behave and how they are formulated.

Bayes' theorem has more flexibility in this regard—a Bayesian prior (and the resultant relationship it describes) can be built over time, as well as over many variables and non-linear relationships between states. I recommend looking into graph representations of Markov chains and Bayesian Networks in order to see this in greater detail—Markov chains are weighted, sometimes cyclic, graphs; while Bayesian Networks are necessarily acyclic, and edges are given probabilistically, in a tabular fashion.

In fact, Markov chains can be described as Bayesian processes, implying Bayes' theorem is a generalisation of how HMMs and Markov chains behave. We will prove this now.

Take our Bayesian inference with evidence, $P(I \mid O)$. We have said this infers a hidden variable of a system given an observation. We have previously stated this to be various things, including connecting a sound to the location it came from. Let's say that we are inferring the value of a hidden variable x given some observation y—in which case it would be wise to rewrite it as

$$P(x \mid y) = \frac{P(y \mid x) \cdot P(x)}{P(y)}.$$

Without loss of generality, we can introduce a temporal context into the inference, such that we are inferring x at time t. This entails also giving y a temporal context, although it is worth stating this could be time-invariant, i.e., we get one observation and it stays constant. We now have the equation

$$P(x_t \mid y_t) = \frac{P(y_t \mid x_t) \cdot P(x_t)}{P(y_t)}.$$

Assume now that we want to iterate over times t to make inferences about x at multiple time-points. This has a few consequences. We will

take a series of measurements in the system, in such a way that we wish to connect each individual inference with a measurement. However, these measurements build, and they are relevant to the dynamics of the system as we are inferring them. They shape our prior, which is built on previous inference.

We can reuse our Markov formalism defined previously to get an idea of what that would look like. If we cared about *all* previous values, but the system was memory-less, we would have the sum over all inferences given their evidence, but not any previous inferences. This yields

$$P(x_t) = \sum^{x_{t-1}} P(x_t \mid x_{t-1}).$$

We must also pay careful attention to how this prior was developed through time. This intuition for the inference of x_t itself came from previous measurements. As a result, due to these iterations, we must 'reintroduce' Bayes' theorem to the prior term. We can achieve this by basing the prior off of all the previously calculated posteriors, or, considering the inference at each time-point given all the previous evidence. This is now

$$P(x_t) = \sum^{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_1, y_2, y_3, \dots y_{t-1}).$$

If we were iterating in such a way as to give rise to the above prior, we would necessarily be considering all model evidence (or measurements) as we build our prior. Following in how we built the prior until t, the posterior at t is

$$P(x_t \mid y_1, y_2, y_3, \dots y_t).$$

Altogether, we have recovered the following:

$$P(x_t \mid y_{1:t}) = \frac{P(y_t \mid x_t) \cdot \sum^{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_{1:t-1})}{P(y_t)}.$$

We can say that $P(y_t)$ is proportional to some normalisation constant Z—the integral across the entire distribution. This will give us our final equation, and as such, we have found our recurrence relation

$$P(x_t \mid y_{1:t}) = Z^{-1} \cdot P(y_t \mid x_t) \cdot \sum^{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_{1:t-1}).$$

□

4.3 Kalman Filters

The Kalman Filter is a specific implementation of the HMM developed and refined by Rudolf E Kálmán in the early 60's, with the immediate effect of improving navigation technology in NASA's *Apollo* Space Programme. It considers the case where the HMM's latent and observed variables come from Gaussian distributions and all transitions are linearly determined. There are two equivalent ways of thinking about it—one is the single prediction and correction step, and the other is a time continuous version arising from iterating the previous process. I'll go over both, with the former appearing first.

We have the following equations:

$$\begin{aligned}\hat{x}_k^- &= F \cdot \hat{x}_{k-1}^+ + B \cdot u_{k-1} \\ \tilde{y}_k &= z_k - H \cdot \hat{x}_k \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k \cdot \tilde{y}.\end{aligned}$$

Now we'll walk through the parts of this equation. $\hat{x}_k^-$ is the prior estimate of x_k. It comes from our perfected posterior estimate of x_{k-1} times F, where F is our transition rule, determining x_{k-1} to x_k. $B \cdot u_{k-1}$ comes from our 'control'—these are simply inputs, such as those in a controlled linear dynamical system. Our next equation gives the error between our true measurement z_k and what our measurement would be if the underlying state were actually $\hat{x}_k^-$. H is the rule that transforms state to measurement, so we are 'matching units' here. If H is known and this error is zero, then we are confident $\hat{x}_k^-$ is correct. If not, then we adjust according to the third equation, where K_k is the Kalman Gain. It simply adjusts our estimate in an appropriate way, so as to get our posterior estimate, $\hat{x}_k^+$.

There is additional complexity in these equations, which has been left out for the purposes of this document—specifically, usage of the covariance matrix P. If one were ever to use this model, I would encourage them to use this basis of understanding to look further into such details.

If we iterate this process over all k, we'll get a slightly different looking form for the Kalman filter. It does the same thing, it simply presents it in a different way. This is the well-known EM algorithm. This essentially places a maximum likelihood estimate onto the Kalman filter estimate. The EM algorithm jointly estimates the parameters of the model of the state, as well as estimates of the states themselves. The E step is a Kalman filter, which uses the current estimates to predict the new states. The M step uses this result in an MLE procedure to obtain the parameter estimates. That's the high level overview.

4.4 Kalman Cognition

Does the brain implement a Kalman filter? There is some amount of controversy to this question. Conceivably this is because now we're trying to put specifics on what the brain does, when we don't exactly know those specifics. This is the question that FEP tries to address—it adds implementation details to Bayesian cognition. In general, when we ask questions like this, we are trying to use our human theories to characterise cognition, with no guarantee that cognition follows the statistical or mathematical formalisms humans actually use.

Metaphysics aside, we can be sure of a few things. We know the brain performs inference in the Bayesian sense, creating hypotheses about hidden variables based on model evidence, in order to connect the observation to the cause. The brain forms inferences about the state of the system it is measuring, perfecting those inferences through time, as it learns from a system. We know this form of carefully curated Bayesian inference is similar in principle to the Kalman filter, as was previously examined.

Certain theories of cognition are not only compatible with Kalman Filtering but are indeed formally equivalent, meaning the mathematical expressions reduce to each other under certain transformations or assumptions. Predictive processing, a model for how the brain handles prediction error and correction of model inference, is one such theory. The foundation for it was developed in 1997 by Rao and Ballard, and they stated the theory formally in 1999. It's similarities to Kalman Filtering can be examined by looking at the equations laid out in following article: *Rao and Ballard, 1997. Dynamic Model of Visual Recognition Predicts Neural Response Properties in the Visual Cortex. Neural Computation, 9(4), 721–763.* Equation 2.3 from this paper is the following:

$$\hat{r}(t+1) = \hat{r}(t) + k_1 \cdot U^T \cdot (I - U\hat{r}(t)) \,.$$

Here, $\hat{r}(t+1)$ and U are (in the language of FEP) some internal states representing an input, while I is a sensory state, or an input. We want to infer the optimal representation (perhaps a firing pattern) that allows us to reconstruct I. So, this equation describes the process by which an optimal neural representation is found for a sensory input. Predictive process thus introduces a neural level implementation of model building. U is a network weight and $\hat{r}(t+1)$ is therefore the prediction, or the estimated best representation at the next time-step.

Compare this with the Kalman filter and the similarity will begin to become clear. In particular, we'll sketch out a proof that these two components

are equivalent to one another.

We'll first take our Kalman state estimate below:

$$\hat{x}_k^- = F \cdot \hat{x}_{k-1}^+ + B \cdot u_{k-1}.$$

We can apply the Markov principle to say that, without loss of generality, the prediction step takes the same form:

$$\hat{x}_{k+1}^- = F \cdot \hat{x}_k^+ + B \cdot u_k.$$

Under certain assumptions our transition matrix may go away, yielding

$$\hat{x}_{k+1}^- = \hat{x}_k^+ + B \cdot u_k.$$

We'll consider the definition of control as being error correction, as in a controlled dynamical system, where the error is the distance from a goal. In such a case, the term for the optimal synaptic weight, $k_1 \cdot U^T \cdot (I - U\hat{r}(t))$, becomes equivalent to our control input, and as such, the following is true when our Kalman estimate relates to optimal neural states:

$$\hat{r}(t+1) = \hat{r}(t) + k_1 \cdot U^T \cdot (I - U\hat{r}(t)) \,.$$

With this best estimate, we can indeed determine exactly 'good' how it is.

What Predictive Processing does is find the error associated with this estimate, and propagate it through the network in order to correct the estimate. This can be modelled exactly as a Kalman filter would adjust its own inference (the form of which we've shown is more or less equivalent).

This proof was rather informal (for my mathematical sensibilities, almost to the extent of insult), but it's all that's necessary to gain an intuition about where Predictive Processing invokes Kalman filtering. For more details I highly recommend the above paper, as well as *Rao and Ballard, 1999. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience, 2(1), 79-87.*

Given all this, predictive processing isn't the whole story, and so neither is Kalman filtering. It couldn't be—as we examined, the Kalman Filter is limited in its inferential capabilities. In the case of non-linear, non-Gaussian data, such as the environments the brain often finds itself in, it needs a more powerful mechanism for inference, like Bayesian Filtering—filtering in this case being recursive adjustment of a learned variable, such as what Kalman derived, or what the brain does in order to learn good inferences. As for Predictive Processing, it is rather a useful (and, so far as we know, correct)

theory—however, it has its limitations. This error signal is poorly understood, and certain proposed implementations like neural backpropagation are controversial, in so far as being deemed impossible by many. The computational complexity of many theoretical implementations is very high, to the point of being intractable, even for the brain. It's a good theory—but there's more to it than just that; so we keep searching.

Work in this field—biological, and more detailed theoretical, implementations of Bayesian Cognition—is currently ongoing, with the matter far from being settled.

5 Ideas about Reward

5.1 Game Theory and Formal Notions of Payoff

While this is not strictly relevant to understanding Week Two content, I think it's a good way of stepping into the world of mathematical quantification of behaviour.

Can we indeed quantify behaviour? Consider a game of chess. There are lots of possible moves, but only a small subset of those moves makes sense. In this way, chess games often follow patterns; in the same way, human behaviour is predictable to within a degree of certainty.

First developed by John von Neumann in 1928 when he published the paper *On the Theory of Games of Strategy*, it did not reach truly widespread usage until the contributions of John F. Nash in his 1951 article *Non-Cooperative Games*. In this paper, Nash defined his 'Nash equilibrium,' and proved that for any game with a finite set of actions, at least one Nash equilibrium must exist.

Nash equilibrium is achieved when a player can no longer increase its own expected payoff by changing its strategy while the other players keep theirs unchanged. This means that in competitive games, when Nash equilibrium is achieved, there can be no greater expected payoff—and thus reward is maximised for each player.

What is payoff? Usually synonymous with reward, it is defined by Game Theorists as a reward driving action. We can model, using Game Theory, how the expected reward and decisions made by players relate to each other, so as to gain insight into decision making. An interesting, if somewhat apocryphal, story is that John Nash originally entitled his work 'Governing Dynamics,' presumably because he knew the import of his work—formal, comprehensive notions of reward would yield the very dynamics governing behaviour and decisions. Indeed, game theory has found its most spectacular

usage in Economics, where it is used to predict how people will behave given a pool of resources (some market). Four Nobel Prizes have been awarded for the application of game theory to Economics.

5.2 Markov Decision Processes

We can build on our previous discussion about Markov chains to introduce our first formal look at reward—the Markov Decision Process (MDP). MDPs are an extension of Markov chains; the difference is the addition of actions and rewards. With no action possible (e.g. the observer must wait for a new observation) and no reward is present, and MDP reduces to a Markov Chain. Decision making can be modelled by a Markov process for two reasons—often we must integrate evidence to guide policy making, and our decisions (or actions) are often based solely on the current state of something, in the sense that though we may gain a sense of what actions are good (just like Markovian inference builds a prior) we often base a decision on the current state of the system we are acting on.

MDPs must not only infer variables to guide action, but also determine what a good action is. We now consider our action in addition to the current state, and gather evidence from measurements to infer the state. In this case we must consider action as a transition, because there is often a change in state based on our actions. Reward, in such a case, may be moving the current, inferred state to a desired state.

An MDP is dependent on four things. First we have the state space S and the action space A. We have the probability of transition respective to action, $P(s, s')$. We can also write this as the probability of a switch to an intended state s_{t+1}, $P(s_{t+1} = s' \mid s_t = s, a_t = a)$, or the probability of $s_{t+1} = s'$ if our state and action equal some particular values. Finally, we consider the reward associated with this action, $R(s, s')$.

Ultimately we aim to find a function for policy making, often denoted by $\pi(s)$, which will give the relationship between current state and best (most rewarding) action. Resultantly, we want $\pi(s)$ to maximise our cumulative reward. This policy will guide the decisions we make. We consider reward as being based on a transition between states, in such a way that we wish to control the system with our actions. In the context of FEP, this transition to s' is desired because it makes s, which doesn't match our internal model, change to s', which does; the associated reward is a small free energy.

The first step will be to define 'reward.' We can't use the payoff defined by Game Theory, as these are not games—there is no competition. We may use the Bellman model for MDP, which emphasises not necessarily

reward but return, which is the cumulative reward associated with each state. Return G is defined as the following:

$$G_t = \sum_{i=0}^{N} \gamma \cdot R_{t+i+1},$$

where our short term reward R for a given action and state is given by :

$$R_{a,s} = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right].$$

Consider the reward as feedback, which occurs after an action is taken and a resultant state transition happens. For generality, we've avoided using a specific definition for R. This recursion holds true no matter what we define R to be.

In our return G, note our variable γ or the discount on our reward. This ensures convergence. Beyond the mathematical utility of it, it gives our agent (a unit in a 'behavioural' simulation who is making decisions) a sense of priorities—should it act for future or immediate reward? Do we want hesitancy or impulsiveness? Anxiety or recklessness? This is a parameter that must be tuned to find an optimum in the middle.

From our state, action, and reward we can calculate a value function which tells us how good a state and a corresponding action would be, or in other words, how good is it to be in a particular state, and how good is it to take a particular action. It informs our agent of how much reward it should expect if it takes a particular action in a particular state. The Bellman equation for state value $V(s)$ and action value $Q(s, a)$ are considered jointly but within separate equations. It is related to our return given our state. It looks like this:

$$V_\pi(s) = \mathbb{E}_\pi\left[G_t \mid S_t = s\right]$$
$$Q_\pi(s, a) = \mathbb{E}_\pi\left[G_t \mid S_t = s, \mathcal{A}_t = a\right].$$

Here, $\mathcal{A}$ is our 'advantage' function. That isn't too important yet. Taking into account our policy, this value function tells us how good it is to be in state S according to our policy.

Once again, we seek to find a policy for choosing action that maximises both our state and action values. To do this, we maximise our return, so we maximise our cumulative reward.

The algorithm for this entails an iterative, two step value update and policy update. There are some variants that combine this into one step, but

it would be best to introduce them separately for clarity. Let's expand our rules to the following:

$$V_\pi(s) = \mathbb{E}\left[R_{t+1} + \gamma \cdot V(s_{t+1}) \mid S_t = s\right].$$

The Bellman equation for value $V(s)$ is related to our reward given our state, and calculates the expected reward at $t+1$ plus the discount of our value for the state at $t+1$, given our current state. What this equation means is the value at $S_t = s$ is the reward we get from transitioning out of that state, plus a discounted average over the possible future states, where the value of each possible future state is multiplied by the probability that we land in it.

We also have for action:

$$Q_\pi(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) \mid S_t = s,\, \mathcal{A} = a\right].$$

We have an expression for the optimal state and action values called Bellman Optimality Equations—they are derived from the following:

$$\pi^* = \arg\max_\pi V_\pi(s) = \arg\max_\pi Q(s,a).$$

An expression for this equation, in terms of V and Q, can be found in the form of Bellman Optimality Equations. I'll leave it to you in case you're interested in enquiring further.

We may end up with something like this for our policy:

$$\pi(s) = \mathbb{E}\left[\sum_0^\infty \gamma \cdot R_{a_t}(s_t, s_{t+1})\right].$$

Breaking this equation down, we will see a number of interesting things. To begin with, our policy function only cares about the expectation $\mathbb{E}$, or most likely value (generally a mean), of the cumulative reward. Variance in our reward is often disregarded. We also see that the reward is based on this transition between states in such a way that we wish to control the system with our actions.

5.3 Reinforcement Learning

What about inference for reward construction? Consider an MDP where certain probabilities or rewards are unknown. Rather than infer our prior to infer our posterior, which would be possible, but would be rather silly, the problem becomes an example of Reinforcement Learning (RL)—in RL,

exploration is incorporated into the problem of reward driven action. This is another reason why we began with MDPs—they bridge inference with reward, so we can finally arrive at reward mediated learning.

RL inherits all the interesting ideas about reward and decision making that MDP has, with additional components that often correspond to real world complexity. Let's begin first with exploration. We are considering a landscape or a problem where the reward for an action is unknown to the agent. We are forced then to sample our landscape so as to form ideas about what the reward associated with certain actions is, or, to explore our options. This is the learning portion of RL—building a model of the environment so as to find rewards associated with certain actions on certain states. Recall the definition of policy π—this learning process give us a model from which we can construct a policy.

We seek to learn a model H such that we can create pairs of different observations and states,

$$H_{t-1} = ((s_1, y_1), \ldots, (s_{t-1}, y_{t-1})).$$

Based on this model, we can reconstruct the likely reward from applying an action to a state by basing our reward off of our observation and inferring the associated state. R_t becomes some function of our observation $R(y_t)$, associated with an inferred s_t. We need to gather enough observations to build this model—this comes from exploration.

In the Second World War, the question of 'how much to explore or exploit' vexed Allied Scientists to such an extent that they proposed sending the question to German Scientists too, as a means of distraction. Luckily, RL gives us a way of looking at exploration formally. We saw so-called 'greedy' and 'ε-greedy' methods in the RL tutorial on Friday, which both performed sub-optimally—greedy methods settle on an action that is not in fact optimal but which appears to work 'well enough,' while ε-greedy methods over-encourage exploration, to the extent of missing maximising reward in the interest of continuing to search. Neither is good. We need an optimum in between exploration to identify something that is actually sufficiently good, and exploitation to take advantage of that when we find it. We defined the solution to this balancing act as 'decaying ε-greedy, which occupies a middle ground between the two.

We need to determine the proper form of learning necessary to integrate evidence appropriately, before our decreasing likelihood of exploration causes us to settle on a sub-optimal solution. In order to do this, we could use Thompson sampling, which samples from the posteriors we create about

rewards given some action taken. So, in a sense, we are using Bayes' theorem to guide learning, meaning the beginning scenario of inferring a prior to infer a posterior wasn't that silly. However, we're doing it in a very specific, guided way, and we're crafting a prior iteratively based on our experience, in an effort to then choose the best future actions. This may remind you of how Markov chains work, and how Bayesian cognition functions.

To begin with, all actions are assumed to have a uniform (or equal at all points) distribution of reward probability. We have no sense of best action or possible reward, so we assume we know nothing by showing no preference. For each observation obtained from an associated action, a reward is generated, and based on the reward a new distribution is generated with probabilities of reward for each possible action. We may get a peak associated with the actions we have taken and no change for actions we haven't tried yet. Further observations are made based on these prior probabilities obtained each round, which then updates our reward distributions. After sufficient observations, each action will have a reward distribution associated with it which can help the player in choosing the actions wisely to get the maximum reward possible.

What this means is we sample our landscape, and as observations are gathered, the distribution is updated according to Bayes' theorem.

Mathematically, there's quite a bit of set up to cover, so we'll move on to the other interesting components of RL for the time being. Here is an extended overview of the mathematics behind decaying ε-greedy and Thompson Sampling: *A Tutorial on Thompson Sampling*.

We have an alternative to 'policy learning,' which estimates our value function as above, by revising our policy that maps action to reward. We can use 'value learning,' which is subtly different, in that it learns a value function without a prior. In fact, because it is learning without forming a model of the environment, we can disregard our memory—we no longer care for what we have learned. We know what a memory-less model that only gives one action for one time-point is—it's an MDP.

Q-learning, an RL algorithm, implements our Bellman equation in the previous section. We perform an action to explore our landscape. A Q-table is created, listing actions and their associated rewards through time, and we build our function $Q(s, a)$ from this. Imagine a grid environment, where the agent is moving towards a goal, or a real life analogue of a mouse working through a maze. The agent will take an action (move left one unit) and find the associated reward. This goes into the Q-table, which records the action at the place and the reward. Similarly, the mouse may turn left, find a dead end, and record the action, the place, and the reward (or lack thereof). We

put this into our Bellman equation as an action mapping to a reward. This requires the same exploration as our policy-learning, but the actual learning itself takes a different form—this is a subtle but important difference. We do this iteratively to craft our Q function. Once this function is estimated properly, we can decide on a likely best series of actions (e.g. the best path to take out of the maze).

The final component of RL is looking at an adaptation of Q-learning, which involves planning. Closer still to what the brain does to learn and make decisions, it combines inference with prediction in a dynamic fashion. In fact, the most common implementation is called Dyna-Q. Planning is done while learning and interacting with the environment, in such a way that determining possible models can help improve a policy.

We noted that Q-learning has no knowledge of our reward functions. It observes or infers s, takes an action, observes or infers s' and the associated reward. It updates the Q function with what the agent has learned.

In Dyna-Q, we also utilise a model of our reward function, and the transition matrix defining the probability of transitioning from s to s' if we take action a. The agent learns this by exploring the landscape, and additionally, the algorithm 'plans' by allowing our agent to simulate the actions it could take—almost as though it were planning in its head. This perfects the model the agent learned by generating information and integrating evidence. The agent performs a model learning step by randomly selecting a previously observed state-action pair, then asking the crafted model about what happens if that action is taken and s moves to s'—what is the associated reward? Using this simulated transition, Dyna-Q updates our reward function by Q-learning, as though it were a real experience derived from exploring the environment. We have our agent try out actions without actually trying them, instead applying our model as we learn it, to determine whether those actions would be good ones *before* they are taken. If we plan ahead effectively, Dyna-Q can drop the exploration needed by a factor of ten—a 90% reduction in the number of actions needed to get to a goal.

Dyna-Q unifies planning, learning, and acting, in a way that is very similar to what the brain does. The paper *Kanai et al, 2019. Information generation as a functional basis of consciousness. Neuroscience of Consciousness, 5(1)* argues that this concept—the generation of information, possibly counterfactual information for the purposes of planning and decision evaluation—is integral to explaining why consciousness has arisen in humans. They mention Dyna-Q specifically when they are building the groundwork to formalise information generation, saying the following:

> 'a model based approach [to learning] allows an agent to adapt to new goals flexibly because it can use the internal model to optimise its behaviour without trial and error... the ability to generate information enables an agent to perform mental simulations for planning future action sequences, which would be otherwise difficult only with a collection of reflexive behaviours.'

Is RL conscious? It may seem nonsensical on the surface, but we may ask a different question that is valuable for understanding the capabilities of our model—does RL have a quality that consciousness also possesses? In maths we often prove the relationships between structures and objects, in such a way that we can say with certainty 'every object X has quality Y; no object A does B; all things that do J are L objects.' This has made it necessary to define the relationships between things, and I would answer the question with an idea from such proofs—information generation is a necessary, but not sufficient, condition for consciousness. One can perform some information generation and not be conscious (it is insufficient), but nothing that can't generate information can also be conscious (it is necessary). Something else (possibly many other things) are needed to constitute a sufficient set of qualities. The authors avoid giving an answer, opening the floor to the debate. Some metaphysicians and panpsychists would disagree rather intensely with me.

As for the latter question, the answer is thus yes, speaking to the promises of RL in neuroscience. We can not only use this to model cognition and connect simulation insights to reward and learning in the brain, but we can define other behavioural rules and evaluate how these impact decision making. A colleague of mine here at Stony Brook has done some interesting work using agent based simulation (with evidence collection rules) to look at the spread of false memories and collective delusions—see the paper: *Luhmann and Rajaram, 2015. Memory transmission in small groups and large networks: An agent-based model. Psychological Science, 26, 1909-1917.* RL has been used to investigate FEP as an effective learning rule—see this paper: *Friston, Daunizeau, and Kiebel, 2009. Reinforcement Learning or Active Inference? PLOS ONE, 4(7).* There are quite a lot of ways to use agent based simulations in psychology and neuroscience, because ultimately, just like humans, agents *behave*.

All this makes RL a compelling new modelling technique for questions in neuroscience—when the field gets around to implementing it.

5.4 Biological Reward-Based Learning

On the subject of neuroscientific ideas of reward, reward based learning isn't confined only to RL—in fact, it began with the brain.

Reward is integral to classical conditioning (a form of associative learning)—reward, in fact, drives conditioning. Learning was Pavlov's main reward function—one set of dogs observed a bell ring and a sausage follow, and another, a bell ring without consequences. The bell begins to predict the sausage in the former group, whereas the latter are decidedly neutral about bells. No action is strictly required, as the sausage happens with or without action, but this is a clear example of how reward mediates learning. In general, in associative learning, a person learns an association between two stimuli only when the events are no longer neutral. Eventually, the neutral stimulus elicits a response on its own, as a result of being paired with a reward. We come to expect a reward when the stimulus is presented, because this is the model we have learned of the world.

In other cases, we know we may exert an influence on whether that stimulus or event arrives, and so reward becomes associated with that action too. Rescorla and Wagner, Yale psychologists, later formalised these ideas with proper mathematical equations. This model is now called Rescorla-Wagner learning.

For a more extended overview of this, I would recommend *Schultz, 2015. Neuronal Reward and Decision Signals: From Theories to Data. Physiological Reviews, 95(3), 853–951.* Otherwise I want to introduce some specific analogues to what we discussed above, beginning with 'is there a relationship between model learning, models of reward, and error control?'

The first thing we may observe is that the error signal associated with learning, adjusting our ideas or representations of the world around us (internal states in FEP, prior models in Bayes' theorem, environmental models in Dyna-Q) has been observed to coincide with dopaminergic neural firing, a neurotransmitter associated with reward. Note my phrasing here: we don't know how error is handled in the brain, and in fact, certain theories like backpropagation are extremely controversial. However, certain experiments present a stimulus which the experimenter knows will cause an error in some model, and then perform scans on the brain; in these experiments, we often observe a dopaminergic signal in the brain, *suggesting* that dopamine and dopaminergic neurones are related to error encoding.

Carrying on with classical conditioning, this makes sense—our model to begin with is that the bell and the sausage are not connected. Nothing in our experience has suggested we are wrong to think so. Then, a bell

rings and a sausage appears. While food is always connected to reward, there is another facet of this that is relevant—our model was wrong. We infer that the bell and the sausage were connected, and we may be sure of it given enough observations of bell and successive sausage. The error in our model coincided with dopaminergic firing. Is it merely a coincidence? Or is there a clever evolutionary explanation for it—correcting a model is rewarded highly, because having a correct model keeps us alive. We can spot the bear, or escape the maze, with a correct model.

The trouble with this is, Bayesian cognition does not explicitly account for reward or error reduction. This should be evident if one were to think about what things like FEP or Predictive Processing are trying to say—the reward isn't reward, strictly, it is minimising surprise. These two ideas aren't necessarily incompatible, but FEP disregards reward entirely by saying everything comes down to minimising surprise. Our reward isn't dopaminergic pleasure or joy—it is matching expectations to reality by moving either one (through learning, or action, respectively). This is complemented by the observation that, if we refer to the paper linked in the above section, Friston has *replaced* reward with free energy in his RL simulations. Broadly, these two theories—Bayesian cognition and reward-based learning—are similar in principle, but the details differ tremendously.

Schultz has another great 2016 article about reward and prediction error, in the non-Bayesian sense, called *Dopamine reward prediction error coding. Dialogues in Clinical Neuroscience, 18(1), 23–32.*

Miller, Kiverstein, and Rietveld have a 2020 article called *Embodying addiction: A predictive processing account. Brain and Cognition, 138* discussing a Bayesian re-evaluation of reward based theories (specifically, the nature of addiction) in cognition.

Does the brain do reinforcement learning? Certainly there are analogies—a model is formed of the environment, learning happens in parallel with planning, and actions are considered to cause transitions to better, or intended, states. Temporal-Difference learning was developed for RL agents as an algorithm that models the expected reward of an action or stimulus given a model, and so there is comparison to be made there. Temporal-Difference learning is descended directly from the Rescorla-Wagner model.

There is some hidden complexity in the brain, though. For one thing, our environments are not static, as in RL—they are fiercely dynamic, with states changing in highly non-linear ways, independently of our actions. People are also not entirely reward oriented—we often do anomalous things or nothing at all, which is in direct conflict with RL. When placed in a dark room, there is no reward associated with taking any action—there is nothing to predict,

and so there's no prediction error. In that case, why don't we stay still, and do nothing? We could say that the reward in this case is exiting the room, and so an RL agent would be motivated to continue exploring and finding optimal solutions to the problem, like perhaps discovering food outside the room—but this is not compatible with the Bayesian idea of prediction error driving action. Moreover, reward is still poorly defined in many scenarios, from a purely quantitative point of view.

Clearly there are a lot of open questions on modelling consciousness. This actually a good thing—after all, if we were so simple as to understand our own consciousness, we probably wouldn't be very conscious at all. If you are interested in this to a greater extent, I recommend the article *Neftci and Averbeck, 2019. Reinforcement learning in artificial and biological systems. Nature machine intelligence, 1, 133-143.*

6 Conclusion

The brain is a statistical machine. It learns how to make the best inferences possible in order to cope with a dynamic, complex environment, and then uses those inferences to learn about and act on the environment it is in. Prior perfection, posterior estimate. There are thus direct analogues in known statistical algorithms, and just like we can model learning, reward, and inference using maths, we can explain how our brain has the power to be cognitive. We have examined these techniques both as they relate to data analysis and as they relate to cognition, so that you might understand both as data driven neuroscientists, and use one to inform the other. This will give you a unique tool set to do high-impact research, of the sort that answers fundamental questions and asks interesting new ones; I truly hope you are as excited as I am about the world of opportunity this affords you.