

# **An Intuitive Introduction to Bayesian Inference and Computational Modelling**

Philipp Schwartenbeck

Centre for Cognitive Neuroscience, Salzburg

Wellcome Trust Centre for Neuroimaging, London

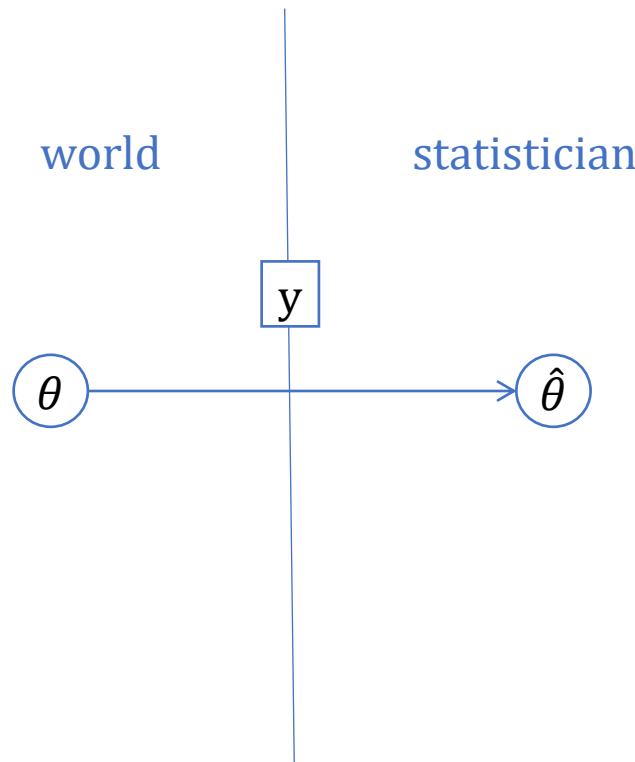
DK+ Winter School, 2017

# What this talk is about..

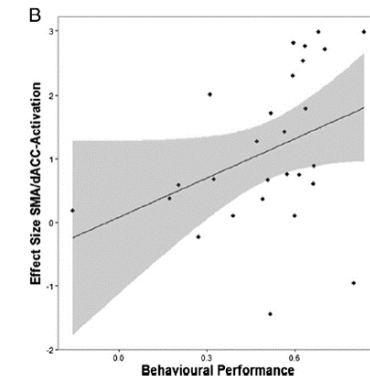
1. In statistics, we try to build models of the world.

True states, e.g.

- Relationship between two variables
- Probability of heads when tossing a coin
- Mean difference between groups



Simple example: Regression



$$y = b_0 + b_1x + \varepsilon$$

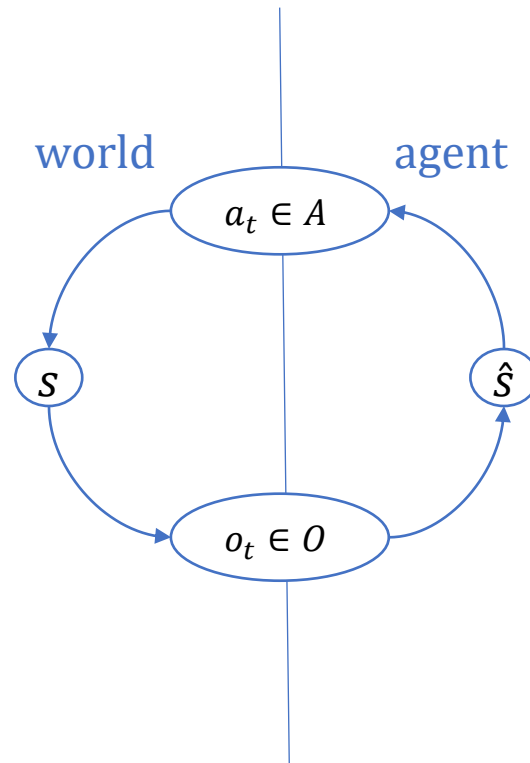
↑                    ↑  
model            error

# What this talk is about..

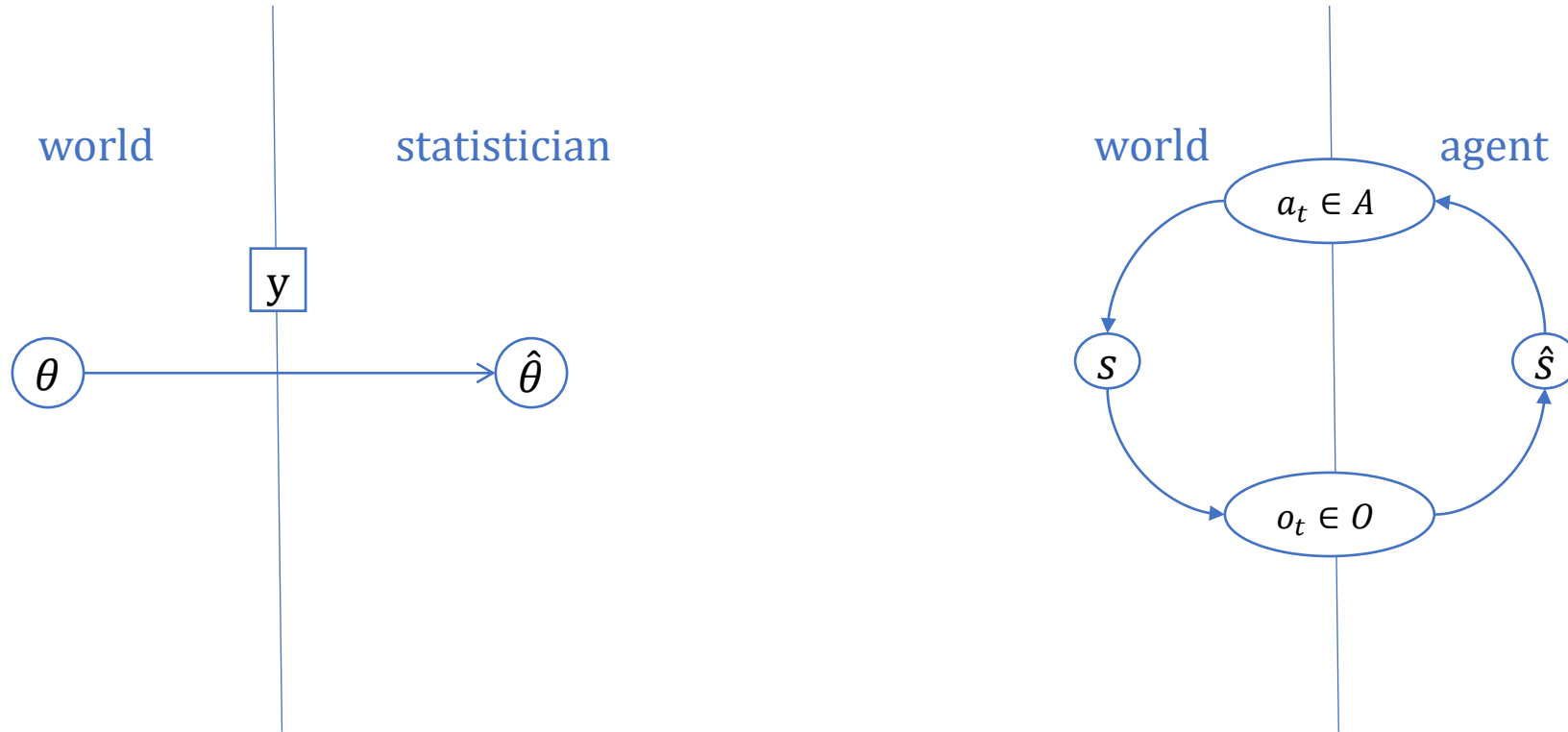
2. In cognition, we try to build models of the world.

True states, e.g.

- A particular object
- Temperature
- A specific context



# What this talk is about..



This talk is about building models, and why Bayes is really helpful in doing so.

# Overview

1. An (intuitive) introduction to Bayesian inference.
2. An (intuitive) introduction to computational modelling.
3. (Bayesian) computational modelling: Parameter estimation.
  - Frequentism vs. Bayes
4. Worked examples:
  - Context inference – hands on!
  - [Computational Psychiatry]

1. An (Intuitive) Introduction to Bayesian Inference.

# Bayes: History



Thomas Bayes, 1701-1761.

Didn't really know what Bayes is all about.



Pierre Simon Laplace, 1749-1867.

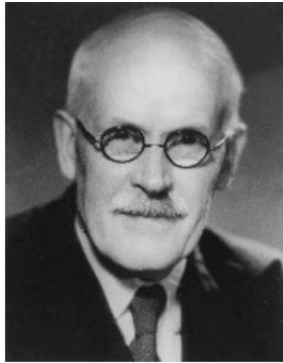
# Bayes: History 1950-now

## Bayesian Statistics

(esp. prior specification and Bayes = logic)



Edwin Jaynes



Harold Jeffreys



Jimmie Savage



Richard Cox

## Bayesian Brains



v. Helmholtz



Alexandre Pouget



Peter Dayan



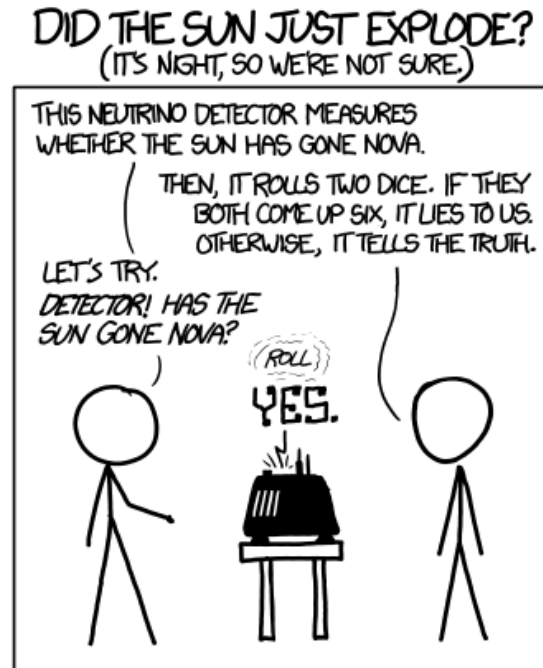
Josh Tenenbaum



Karl Friston



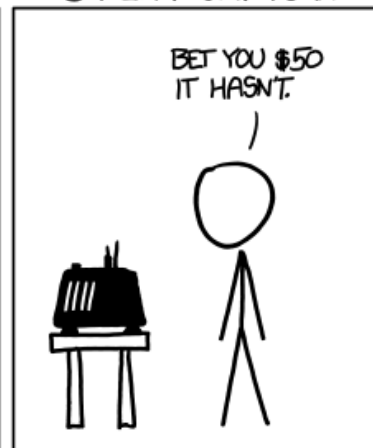
# Bayes: What's the point?



FREQUENTIST STATISTICIAN:



BAYESIAN STATISTICIAN:



# Bayes: What's the point?

Bayes' theorem is a simple result from probability theory.

Bayesian inference specifies the optimal way to account for uncertainty.

- Optimal really means *optimal*, cf. Cox's theorem

Bayesian inference provides an account of how one should integrate information, e.g.,

- Expectations vs. data
- Two sources of info (such as vision and touch)

Bayesian agents are very good at performing learning and inference.

# Why is Bayes useful for neuroscientists?

Brains are information processing systems.

Bayes explains how different sources of information can be integrated under uncertainty.

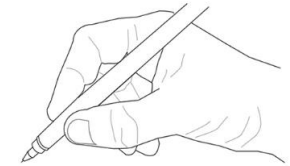
- If we would create an artificial information processing system, making it a Bayesian system would be a good idea

Bayesian systems create interesting hypotheses about our brains

- Deviations from Bayes provide interesting insights into limitations and approximations

# Bayes: a simple example

Estimating the width of a pen when trying to grasp it

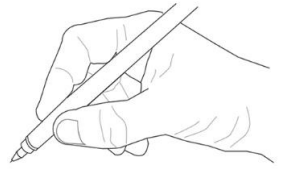


Bayesian systems represent information as probability density functions (PDFs)



Importantly, PDFs represent both the ‘best estimate’ and the uncertainty in that estimate

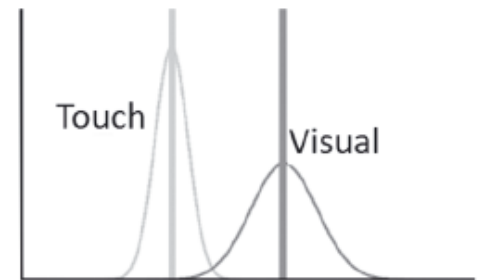
# Bayes: a simple example



Imagine you have access to visual and tactile information about pen width, i.e.

$$\frac{p(\text{width}|\text{vision})}{p(\text{width}|\text{touch})}$$

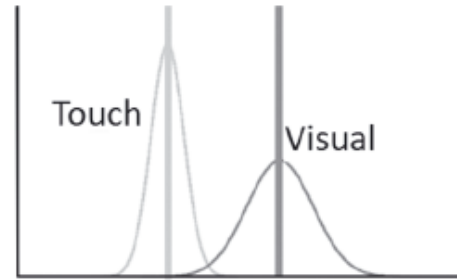
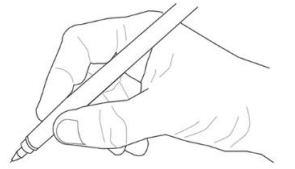
'|'  $\Leftrightarrow$  “given” (conditional probability, see later)



What is the best way to integrate  $p(\text{width}|\text{vision})$  and  $p(\text{width}|\text{touch})$ ?

- Bayes' rule!

# Bayes: a simple example



What is the best way to integrate  $p(\text{width}|\text{vision})$  and  $p(\text{width}|\text{touch})$ ?

$$p(\text{width}|\text{vision}, \text{touch}) \propto p(\text{vision}, \text{touch}|\text{width}) \times p(\text{width})$$

Posterior density

Likelihood function

Prior density

# Bayes: a simple example



$$p(\textit{width}|\textit{vision}, \textit{touch}) \propto p(\textit{vision}, \textit{touch}|\textit{width}) \times p(\textit{width})$$

$p(\textit{width})$  reflects *prior knowledge*

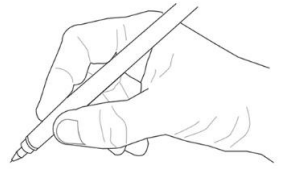
- E.g., from dealing with pens previously
- Ignore for now and assume constant

$p(\textit{vision}, \textit{touch}|\textit{width})$  reflecting *likelihood* of sensations given true width

- Assume independent noise:

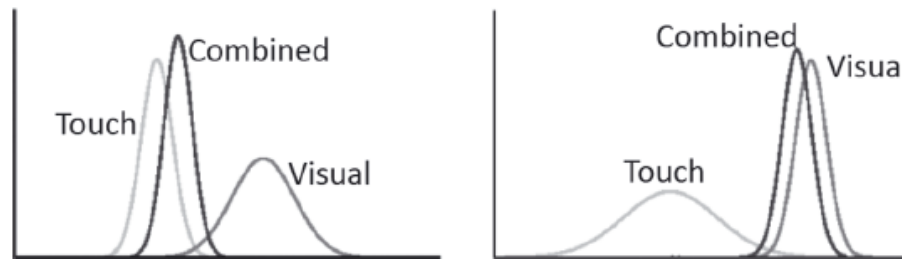
$$p(\textit{vision}, \textit{touch}|\textit{width}) = p(\textit{vision}|\textit{width}) \times p(\textit{touch}|\textit{width})$$

# Bayes: a simple example



$$p(\text{width}|\text{vision}, \text{touch}) \propto p(\text{vision}, \text{touch}|\text{width}) \times p(\text{width})$$

A Bayesian system will combine info according to the precision of the estimates:

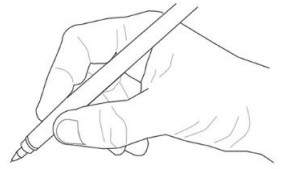


The width of the combination is smaller than the widths of the two other distributions!

- Posterior has on average lower variance and entropy than prior (Felsenstein, 2008)



# Bayes: a simple example



$$p(\textit{width}|\textit{vision}, \textit{touch}) \propto p(\textit{vision}, \textit{touch}|\textit{width}) \times p(\textit{width})$$

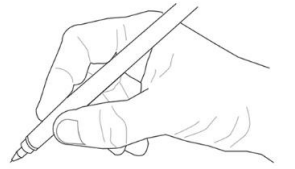
So far:

Bayes provides a measure of uncertainty about measure of pen width

Bayes provides optimal (rational) way of combining sources of information

- Weighted by the relative uncertainty

# Bayes: prior knowledge

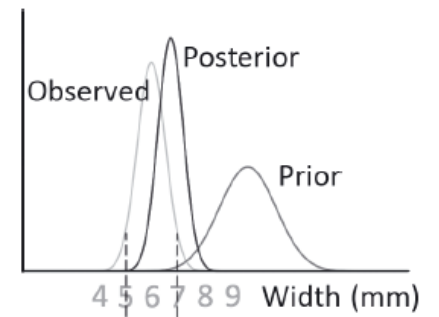


Just as in combination of two different sources, Bayes provides account of ‘optimal’ way to integrate prior knowledge:

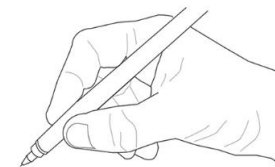
$$p(\text{width}|\text{observation}) \propto p(\text{observation}|\text{width}) \times p(\text{width})$$

$p(\text{width})$  is prior, e.g. knowledge about average pen width from previous examples

- Posterior estimate of the prior biased towards prior mean



# Bayes: prior knowledge

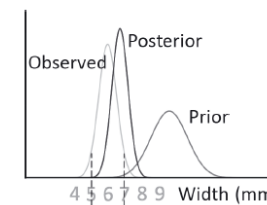


$$p(\text{width}|\text{observation}) \propto p(\text{observation}|\text{width}) \times p(\text{width})$$

What is the functional role of the prior?

“Note that, although the prior might appear to ‘bias’ perception away from the observed value, the effect of applying a prior is in general to **increase accuracy**.”  
(O’Reilly, Jbabdi, Behrens, 2012)

- Because there is uncertainty in the observation



The role of the prior is also to avoid inference being too close to the data

- i.e., to **prevent overfitting**
- Necessary for generalising models

# Bayes: prior knowledge

The prior is the most controversial aspect of Bayesian inference.

- Where do priors come from?
- Can we always control the influence of the prior?

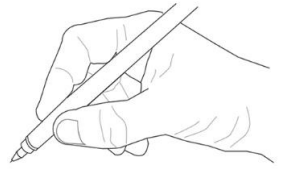
Not using a prior requires as much justification as including a prior.

Using a non-informative prior requires as much justification as using an informative prior.

If it were possible to define objective, formal rules for specifying the prior, Bayes would always be the best choice.

- Particularly, if we could always specify uninformative priors, Bayes would be identical (or at least very similar) to frequentist statistics (shown later)
- Sadly, how to specify a prior can be a problem.

# Bayes: prior knowledge



Together,  $p(\textit{observation}|\textit{width}) \times p(\textit{width})$  specify a model of the (pen-) world

- Called **generative model**, because it allows to generate data and make predictions
- Explicitly model the structure of the world

State space has to be represented explicitly

- Pro: Efficient learning and updating
- Con: Not always known
- Con: possible high dimensionality, e.g. estimate width, colour and weight

# What can Bayes tell us about the brain?

Assumption: Brain represents information as probability functions

Bayes is the best way to integrate information under uncertainty

- Thus, evolution might have brought humans and other animals at least close to being Bayesian systems

Bayes as a 'normative model' – how brain should behave if it were optimal

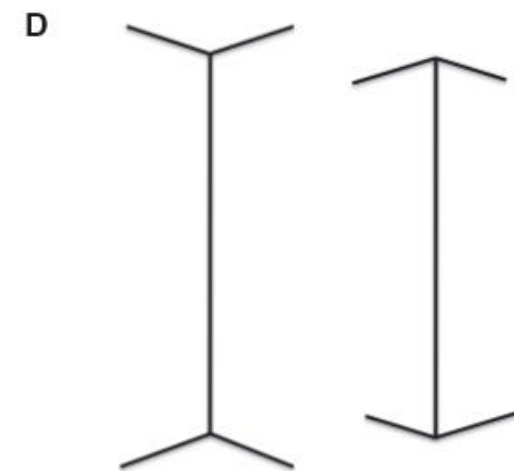
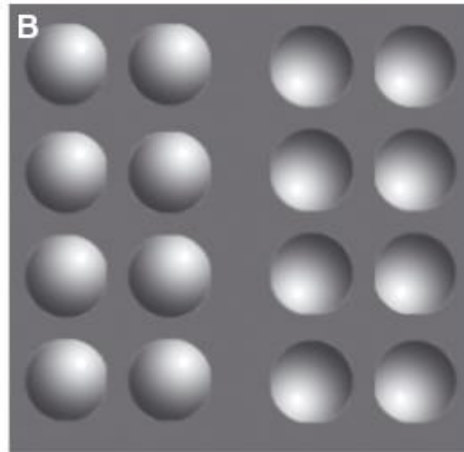
- Derive specific hypotheses for information processes
- Test for specific limitations/approximations

# Bayesian Brains!

Multisensory integration: humans very close to Bayes optimality

- However, some studies show preference for visual information (i.e., a preference for some sensory modalities)

Perception, e.g. visual Illusions as Bayes optimal perception:

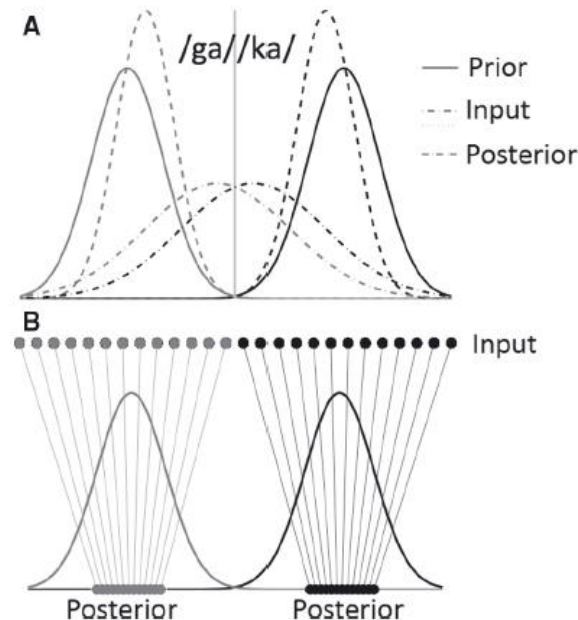


- Some of these priors might be innate

# Bayesian Brains!

Categorical perception of two phonemes that cross a category boundary (such as /ga/ and /ka/) due to Bayesian inference

- Much easier that distinguishing phonemes that fall within a category (/ga/ and another, slightly different / ga / ).





# Bayesian Brains!

Bayesian models can be helpful in informing models of cognition

Example: Attention

- Attention = spatial focus or ‘spotlight’
- Attention = processing one dimension of a stimulus (e.g., colour but not shape)
- Are visual selective and feature-based attention computationally similar?

Bayesian model:

- Spatial focus as manipulation of prior
- Stimulus as reduction in dimensionality of parameter space (‘marginalisation’)

Visual selective attention involves setting up a prior across all dimensions in state space

- E.g. some locations in space are more likely than others

Feature-based attention is about selecting which dimensions of state space to process at all

Reasonable to expect these two computational processes to have different neural implementations

# Anti-Bayesian Brains!

Bayes can be extremely costly (almost always)

- Maintain representation of entire joint probability space
- Update this distribution jointly for all parameters in the model

Bayesian algorithms require complex integrations be computed over these high dimensional spaces

- Hampered uptake of Bayesian methods in science until advent of Monte-Carlo approaches (Gamerman, 1998).

On the other hand, brains are massively parallelized computers

- Some tasks that are complex for digital computers may be easy for brains and vice versa

# Anti-Bayesian Brains!

Non-Bayesian information processing may be favoured in situations in which the Bayesian approach would offer little extra benefit

Example: environment is changing very rapidly

- Bayesian solution almost identical to the simplest possible strategy – to simply act according to the last trial (Summerfield et al., 2011)

Heuristic strategies seem to be preferred in extremely volatile conditions

- Tendency to revert to Bayesian solution when conditions become more stable

# Anti-Bayesian Brains!

“The complexity of the cognitive system makes powerful metaphors such as the probabilistic mind and the Bayesian brain appealing on the one hand, but limited on the other”

- “One problem confronting the notion of the probabilistic mind [...] is the apparent intractability of rational probabilistic calculation”

“[...] much of human inductive inference [is] relying on an adaptive toolbox of simple heuristics”

- “In this view, organisms do not optimize but satisfy”

# Anti-Bayesian Brains!

Often, it is also very difficult to test how we deviate from optimality

- Almost every process is hierarchical and has multiple stages
- Usually, we can only observe output of final stage (Daunizeau et al., 2010)

Cf., hierarchical Bayes and empirical priors (-> HGF)

Subjects may not be rational

- If they are not, inferences about their beliefs may be incorrect
- Explanatory vs. descriptive level

# Bayesian or anti-Bayesian brains?

Bayesian reasoning is optimal and rational

- Combining sources of information according to uncertainty or the precision of these sources

If evolution made us good information processors, we should share some aspects with Bayesian systems.

Relevant question is not necessarily Bayes vs. not-Bayes, rather use Bayes to construct interesting hypotheses

- How/where we deviate from Bayes optimal processing
- How brains might perform approximate Bayesian inference

# Measuring Bayesian Brains

If behaviour is Bayesian, brain must have some way of doing Bayes

- Representing multidimensional PDFs
- Combining these and producing conditional and marginal probability distributions from joint distributions

Simple way to do this would be for neurons to represent parameters of probability distribution

- E.g. mean and variance
- Not necessarily the most efficient way (brain as massively parallel computer)

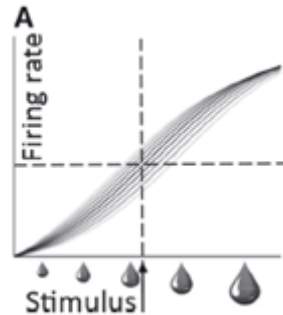
# Measuring Bayesian Brains: Global Population Activity

Population of neurons may all respond in a similar way to value of some parameter

- Neurons will have a range of firing rates due to noise

Average activity of entire neural population proportional to the expected value of parameter

- Hence in fMRI signal we would expect larger signal from neuronal population for larger parameter

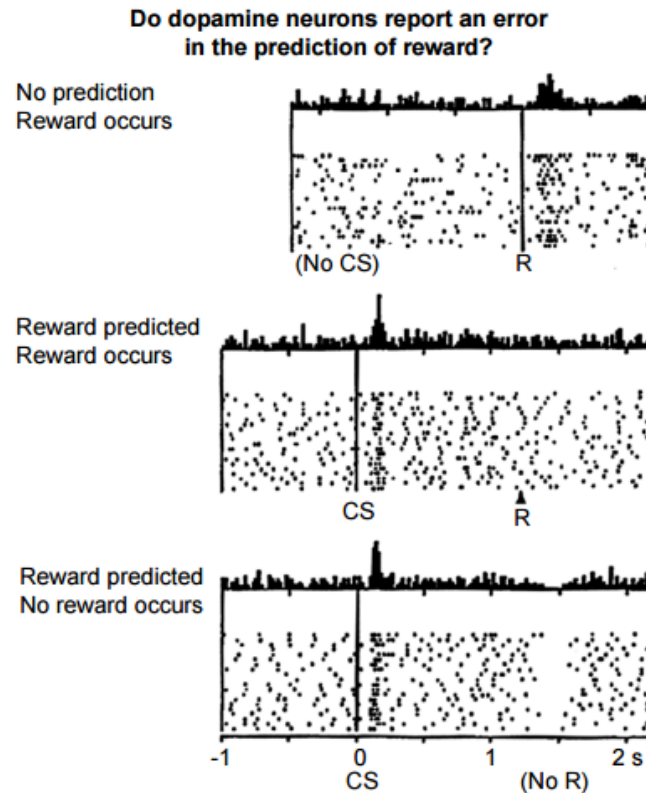


But: difficult to show that variance of firing rates across population is proportional to uncertainty



# Measuring Bayesian Brains: Global Population Activity

Classical example: activity of dopamine neurons reflects reward prediction error (*Schulz, Dayan & Montague, 1997*)

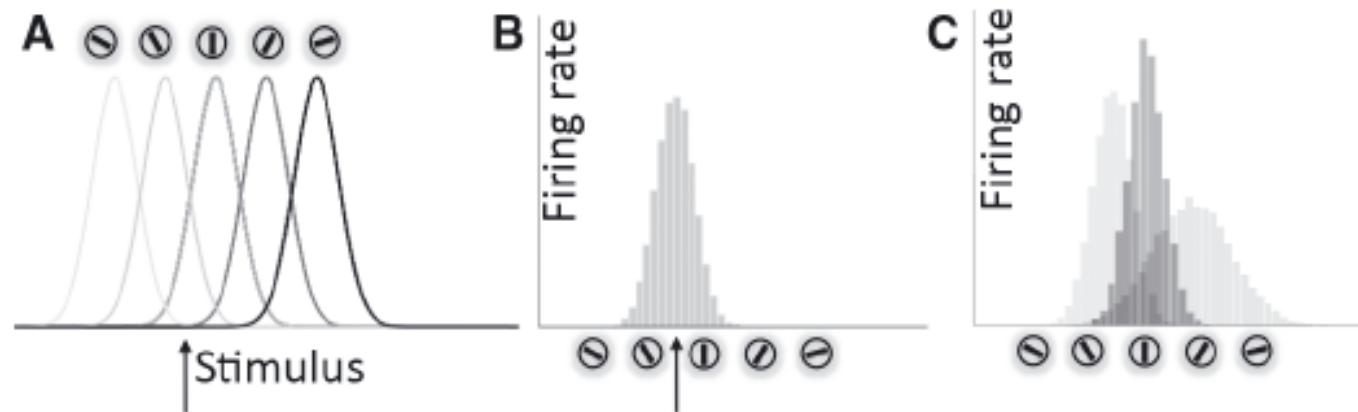


# Measuring Bayesian Brains: Distribution of activity within population

Alternative way is for each neuron to be tuned to respond maximally when parameter takes particular value

E.g., each cell has a tuning curve for orientation in primary visual cortex

- Each neuron is activated in proportion to how close observed stimulus is to its preferred stimulus



# Measuring Bayesian Brains: Distribution of activity within population

Cells may also be tuned to more abstract parameters

- E.g. timescale
- Allowing them to support Bayesian learning algorithms

Difficult to predict reflection in signals based on more global methods (e.g., fMRI)

- Only read out average activity of the whole population of neurons

Averaged population activity need not be related to parameter value at all

- Redistributing activity amongst cells within population
- But alternative methods, such as multivariate decoding, might be option

But: Imaging studies do suggest that global activity increases with increased stimulus specificity

# Interim summary

Bayes provides optimal (rational) way of combining sources of information

- Weighted by the relative uncertainty

Bayes provides both 'best estimate' and uncertainty of our estimate

- E.g., pen width and uncertainty about pen width

Allows for integration of prior information

- Most critical issue

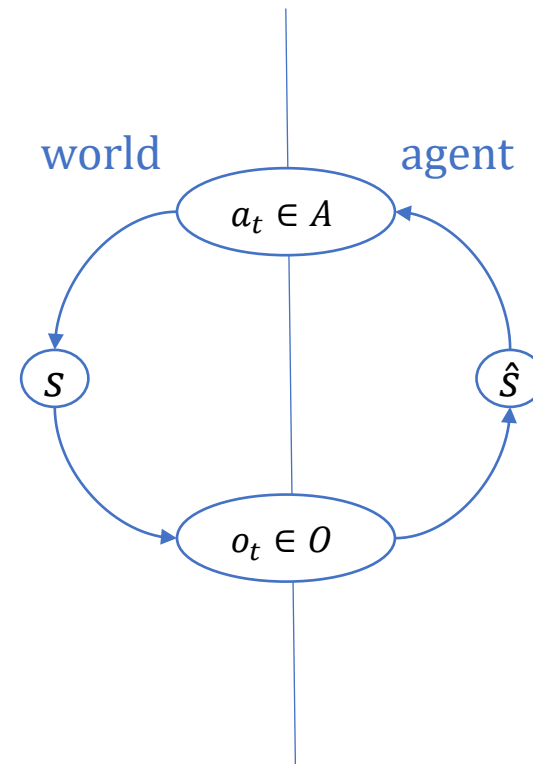
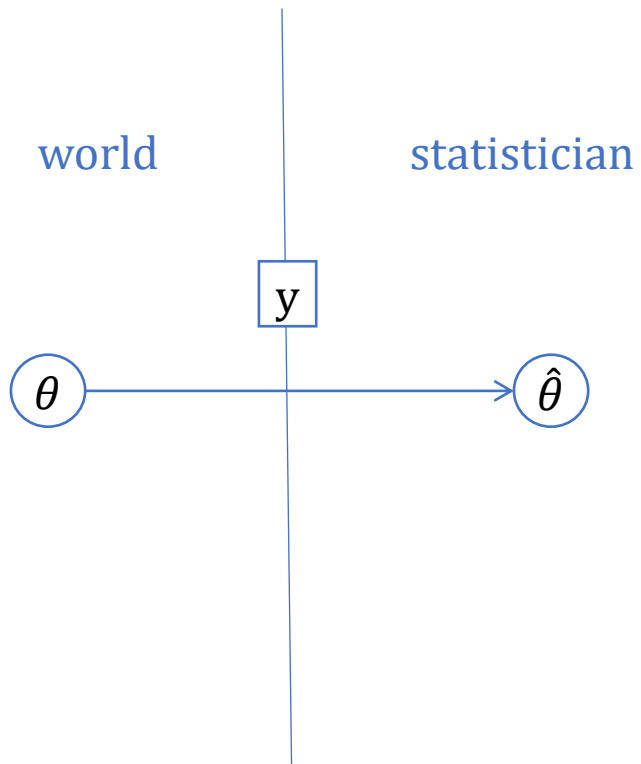
Particular computations might be implemented in different neural architectures

- Important for determining measures of neuronal activity
- Neural correlate really a representation of a parameter, or index of particular neural mechanism?

## 2. An (Intuitive) Introduction to Computational Modelling.

# Why computational modelling?

Both scientists and brains build models of the world.



# Why computational modelling?

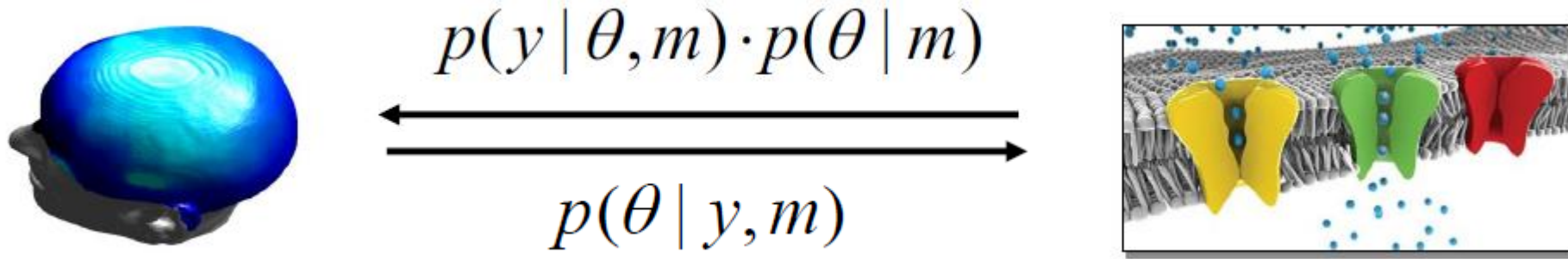
## Computational neuroscience:

- Using computational models to inform our understanding of brain function and dysfunction.

## Computational Psychiatry:

- Focus on the dysfunction part.

# Generative model



Enforces mechanistic thinking: how could the data have been caused?

Simulate ('generate') data  $y$  – can model explain certain phenomena at all?

Inference about parameters via  $p(\theta | y, m)$



# Computational models

Models must necessarily neglect many aspects of reality

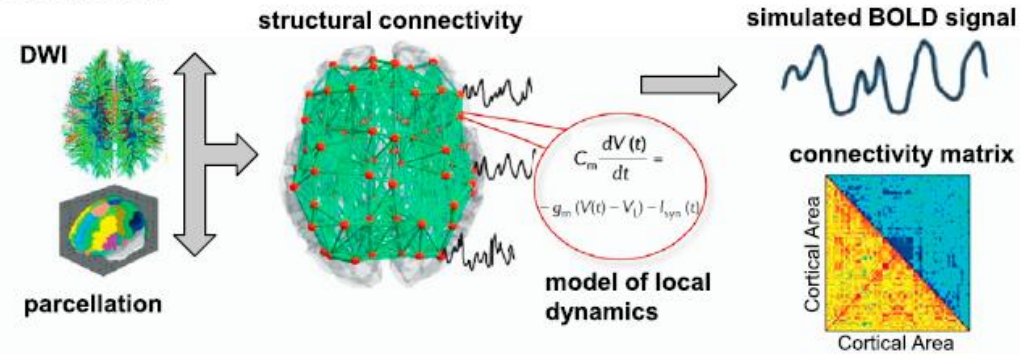
- Represent deeper causal structure of reality
- Mapping between model and reality

Computational (clinical) neuroscience is not a homogenous field: differences in

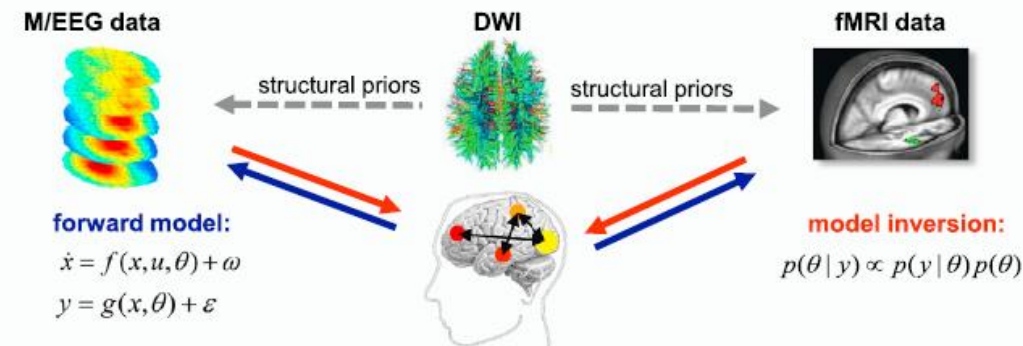
- Intended purpose
- Mathematical techniques employed
- Level of explanation they seek

# Intended purpose of models

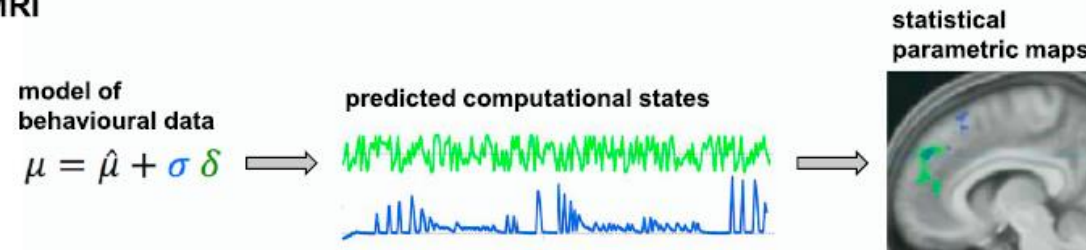
## A Biophysical network models



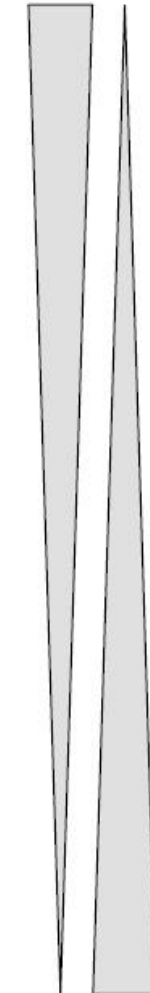
## B Generative Models



## C Model-based fMRI



Biological realism



Estimability

# Mathematical techniques

Most importantly (here): Bayes vs. frequentist

Frequentist: Maximum likelihood

Bayesian: Maximum a-posteriori

- Extension of maximum likelihood

Approximation techniques for Bayesian models

- Sampling (MCMC)
- Variational Bayes

# Level of explanation: normative vs. process models

## **Normative models:**

- Specify what an agent should do, i.e. the '**objective function**'
  - E.g., maximise reward, minimise free energy
- Derive predictions for behaviour and/or brain function from these central principles
- Higher level cognitive function

## **Process models:**

- Provide a model of the mechanistic neuronal implementation of a particular process
- Aims at explaining as opposed to describing a process

# Why computational modelling?

There are at least three benefits from applying computational models to neuroscience and psychiatry (*Teufel & Fletcher, 2016*):

1. Enforcing rigour and precision in the formalization of conceptual models
2. Inspiring useful new conceptualizations of known phenomena
  - Identify core principles of brain processes and disorders
3. Bridging the gap between different levels of explanation
  - From basic neurobiology to conscious experience/suffering
  - Cf., Marr (1982)

# Why computational modelling?

“Conceptual models that are phrased exclusively in linguistic terms inevitably carry a certain amount of vagueness and ambiguity” (*Teufel & Fletcher, 2016*)

Yes – but the true meaning of a parameter or what variance it explains can also be ambiguous!

Still, computational modelling might help to uncover implicit assumptions

- Would remain unnoticed without such formalisations

# Challenges

Danger that models obscure rather than clarify

- They may form basis for assertions that are mutable and inexact

E.g., Bayesian explanations of hallucinations in Schizophrenia:

1. Increased weighting of prior expectation in perception
  - Expectations generate inaccurate percepts
2. Reduction in relative weighting of prior expectation
  - Relatively stronger bottom-up signal that generates aberrant percept
3. Circularity in inferential processing, lack of inhibitory control...

Challenge is to avoid overburdening a model with interpretations that it cannot dissociate

# Interim summary: vital aspects of modelling

Specify purpose of model and its role in the explanatory process

- Level of explanation at which model represents reality

Think about mapping between mental/neurobiological process and theoretical model

- Mathematical concepts need to be carefully chosen
- Capture those aspects of reality that are essential to chosen purpose of model

It may be profoundly useful and informative for a model to break.

“Foremost, we must repeatedly ask ourselves, what are we modelling, how do our model components relate to reality and, crucially, what are we leaving out?” (*Teufel & Fletcher, 2016*)



### 3. (Bayesian) computational modelling: Parameter estimation.

Frequentism vs. Bayes

# Frequentism or Bayesianism?

Falsification vs. model evidence

Frequentism: likelihood of the data given null hypothesis

- Allows to reject the null hypothesis

Bayes: probability of a parameter given the data

- Allows to reject or accept the null hypothesis
- Simple way: Bayes factor (comparing the model evidence for null and alternative)

$B$	$p(m_1 y)$	Evidence
1 to 3	50-75%	weak
3 to 20	75-95%	positive
20 to 150	95-99%	strong
$\geq 150$	$\geq 99\%$	Very strong

# Parameter estimation

Can be Bayesian or frequentist

aka 'model inversion'

- Of Bayesian or non-Bayesian models

Aim: obtain individual parameters that describe data

- Statistics: measure of effect size, mean difference, regression coefficient, ...
- Brain: visual object, current context, width of a pen, ...
- Behavioural modelling: individual parameters that describe subjects' behaviour

# Probability theory basics: conditional probability

## Definition

In a probability space  $(\Omega, \mathcal{A}, \mathcal{P})$  with sets  $A, B \in \mathcal{A}$  and  $\mathcal{P}(B) \neq 0$ , the **probability of A conditioned on B** is defined as:

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(A \cap B)}{\mathcal{P}(B)}$$

# Probability theory basics: conditional probability

E.g., what is the **sensitivity** and **false alarm rate** of a clinical test?

Sensitivity:

$$p(\textit{positive test}|\textit{disease}) = 0.99$$

Probability for false alarm:

$$p(\textit{positive test}|\textit{no disease}) = 0.05$$

# Probability theory basics: product and sum rule

## Definition

In a probability space  $(\Omega, \mathcal{A}, \mathcal{P})$  with sets  $A, B \in \mathcal{A}$ , the **product rule** states that multiplying a conditional probability with the **marginal likelihood** of the conditioning variable yields the **joint distribution** of the two variables:

$$\begin{aligned}\mathcal{P}(A \cap B) &= \mathcal{P}(A|B) \cdot \mathcal{P}(B) \\ &= \mathcal{P}(B|A) \cdot \mathcal{P}(A) \\ &= \mathcal{P}(B \cap A)\end{aligned} \quad \Leftrightarrow \quad \mathcal{P}(A|B) = \frac{\mathcal{P}(A \cap B)}{\mathcal{P}(B)}$$

If  $B_1, B_2, \dots$  denotes a measurable partition of  $B$ , then the **sum rule** states that if we sum over all instances of the conditional variable, we are left with the marginal likelihood of the conditioned variable:

$$\mathcal{P}(A) = \sum_{i=1}^{\infty} \mathcal{P}(A \cap B_i)$$

Applying product rule, it follows that

$$\mathcal{P}(A) = \sum_{i=1}^{\infty} \mathcal{P}(A \cap B_i) = \sum_{i=1}^{\infty} \mathcal{P}(A|B_i) \mathcal{P}(B_i).$$

# Probability theory basics: sum and product rule

E.g., what is the probability of receiving a positive test result in general?

$$p(\text{positive test}|\text{disease}) = 0.99$$

$$p(\text{positive test}|\text{no disease}) = 0.05$$

$$p(\text{disease}) = 0.01$$

$$p(\text{no disease}) = 1 - p(\text{disease}) = 0.99$$

Then: 
$$p(A) = \sum_{i=1}^{\infty} p(A|B_i)p(B_i)$$

$$\begin{aligned} p(\text{positive test}) &= p(\text{positive test}|\text{disease}) \cdot p(\text{disease}) + p(\text{positive test}|\text{no disease}) \cdot p(\text{no disease}) \\ &= 0.99 \cdot 0.01 + 0.05 \cdot 0.99 \\ &= 0.0594 \end{aligned}$$

# Probability theory basics: Bayes' rule

## **Theorem**

In a probability space  $(\Omega, \mathcal{A}, \mathcal{P})$  with sets  $A, B \in \mathcal{A}$  and  $\mathcal{P}(B) \neq 0$ , the probability of A conditioned on B can be written as:

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(B|A) \cdot \mathcal{P}(A)}{\mathcal{P}(B)}$$

This is called **Bayes' rule**.



# Probability theory basics: Bayes' rule

E.g., what is the probability of having a rare disease given a positive test?

$$p(\text{positive test}|\text{disease}) = 0.99$$

$$p(\text{positive test}|\text{no disease}) = 0.05$$

$$p(\text{disease}) = 0.01$$

$$p(\text{no disease}) = 1 - p(\text{disease}) = 0.99$$

$$p(\text{positive test}) = 0.0594$$

Then: 
$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

$$p(\text{disease}|\text{positive test}) = \frac{p(\text{positive test}|\text{disease}) \cdot p(\text{disease})}{p(\text{positive test})} = \frac{0.99 \cdot 0.01}{0.0594} = 0.16$$

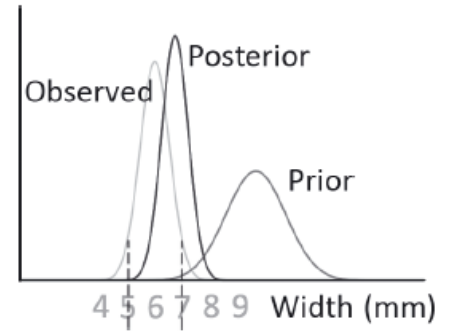
# Probability theory basics: Bayes' theorem

Bayes' theorem more general:

$$p(\theta|y) = \frac{p(y|\theta) \cdot p(\theta)}{p(y)}$$

$p(\theta)$  **prior density** reflects existing a-priori knowledge about  $\theta$

- 'Regularising' the likelihood



$p(y|\theta)$  **likelihood** reflects the probability of observing the data under given  $\theta$

- Without any prior knowledge posterior = likelihood

$p(y)$  serves as a **normalisation factor** to ensure that the posterior is a prob. distribution

$$\int p(y|\theta) \cdot p(\theta) d\theta = p(y) \quad \text{or} \quad \sum p(y|\theta) \cdot p(\theta) = p(y)$$

$p(\theta|y)$  **posterior** reflects update of the prior distribution after making an observation

- On average smaller variance and entropy than prior density (Felsenstein, 2008).

# Frequentist parameter estimation: Maximum-Likelihood (MLE)

For maximum likelihood parameter estimation, we try to find the parameter value that maximises the likelihood function.

Formally:

$$\hat{\theta}_{MLE}(y) = \operatorname{argmax}_{\theta \in \Theta} p(y|\theta)$$

# Bayesian parameter estimation: Maximum a-posteriori (MAP)

For maximum likelihood parameter estimation, we try to find the parameter value that maximises the posterior.

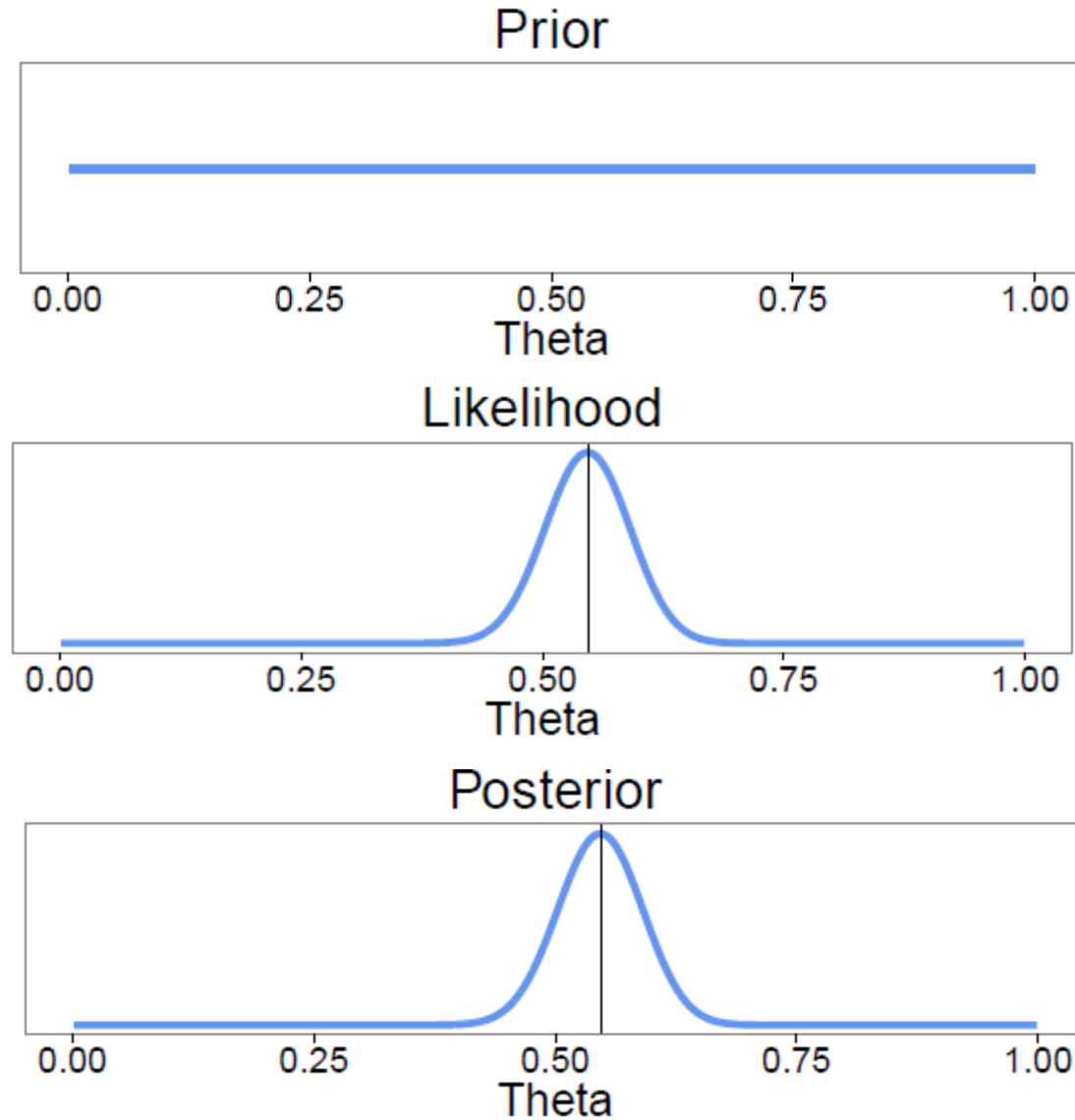
Formally:

$$\hat{\theta}_{MAP}(y) := \operatorname{argmax}_{\theta \in \Theta} p(\theta|y) = \operatorname{argmax}_{\theta \in \Theta} p(y|\theta) \cdot p(\theta)$$

# Relationship between MLE and MAP

Simplest example:

- Estimate probability of success
- $\theta \in [0,1]$
- Tossing a coin, firing of AP, ....



$p(\theta)$

Bernoulli likelihood function  
 $p(y|\theta)$

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

Figure 1. Results of simulating a random sample in a coin toss experiment using a Bernoulli likelihood under a true  $\theta$  of 0.5 and a sample size of 128.

# Relationship between MLE and MAP

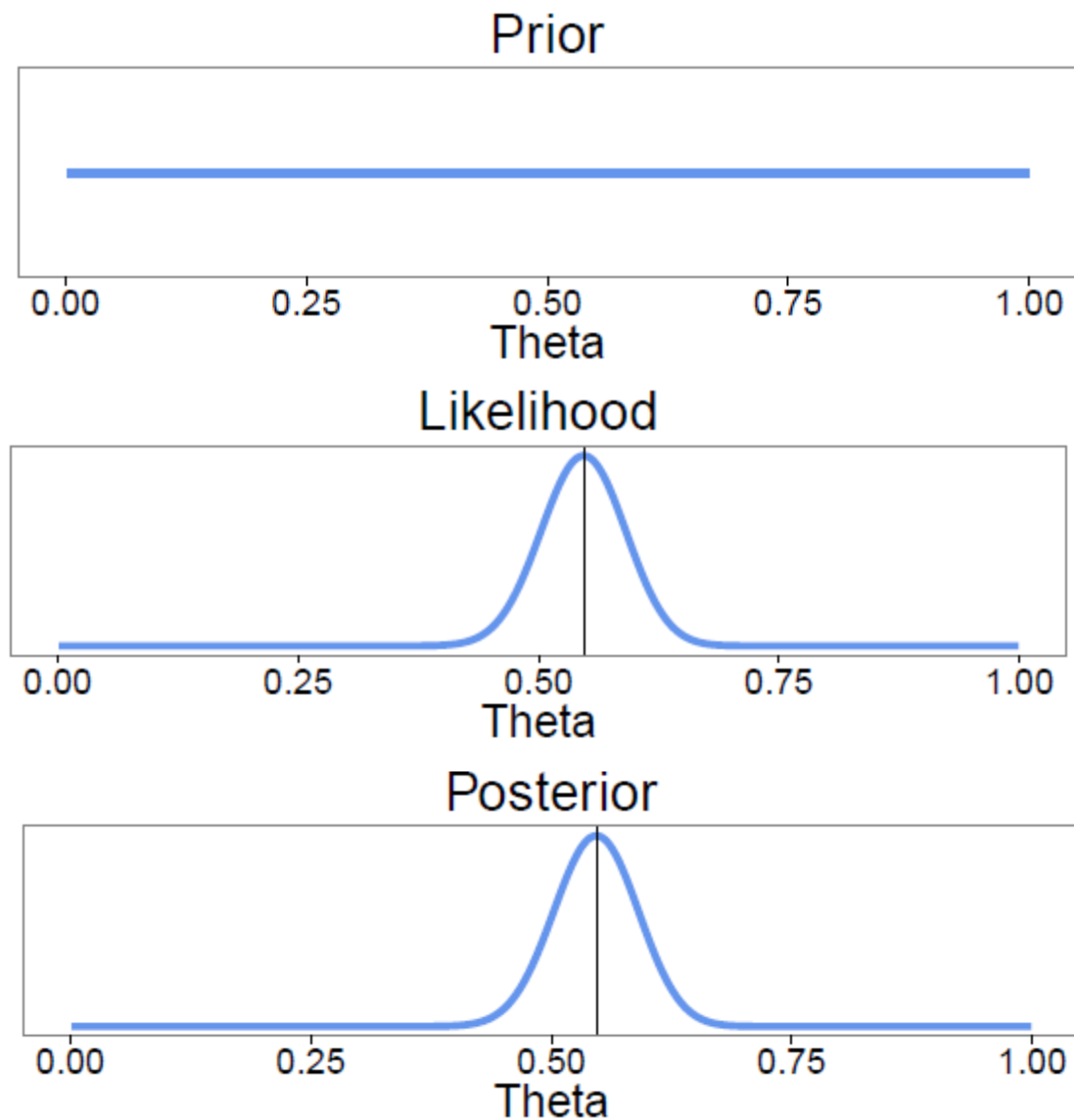
Since the logarithm is a monotonic function, finding the maximum of  $p(\theta|y)$  is the same as finding the maximum of  $\ln p(\theta|y)$ .

Therefore:

$$\begin{aligned}\hat{\theta}_{MAP}(y) &:= \operatorname{argmax}_{\theta \in \Theta} p(\theta|y) \\ &= \operatorname{argmax}_{\theta \in \Theta} \ln p(\theta|y) \\ &= \operatorname{argmax}_{\theta \in \Theta} \ln(p(y|\theta) \cdot p(\theta)) \\ &= \operatorname{argmax}_{\theta \in \Theta} (\ln p(y|\theta) + \ln p(\theta))\end{aligned}$$

This is important, because it means that Bayes and a frequentist solution will agree perfectly if we use a uniform prior.

# Relationship between MLE and MAP



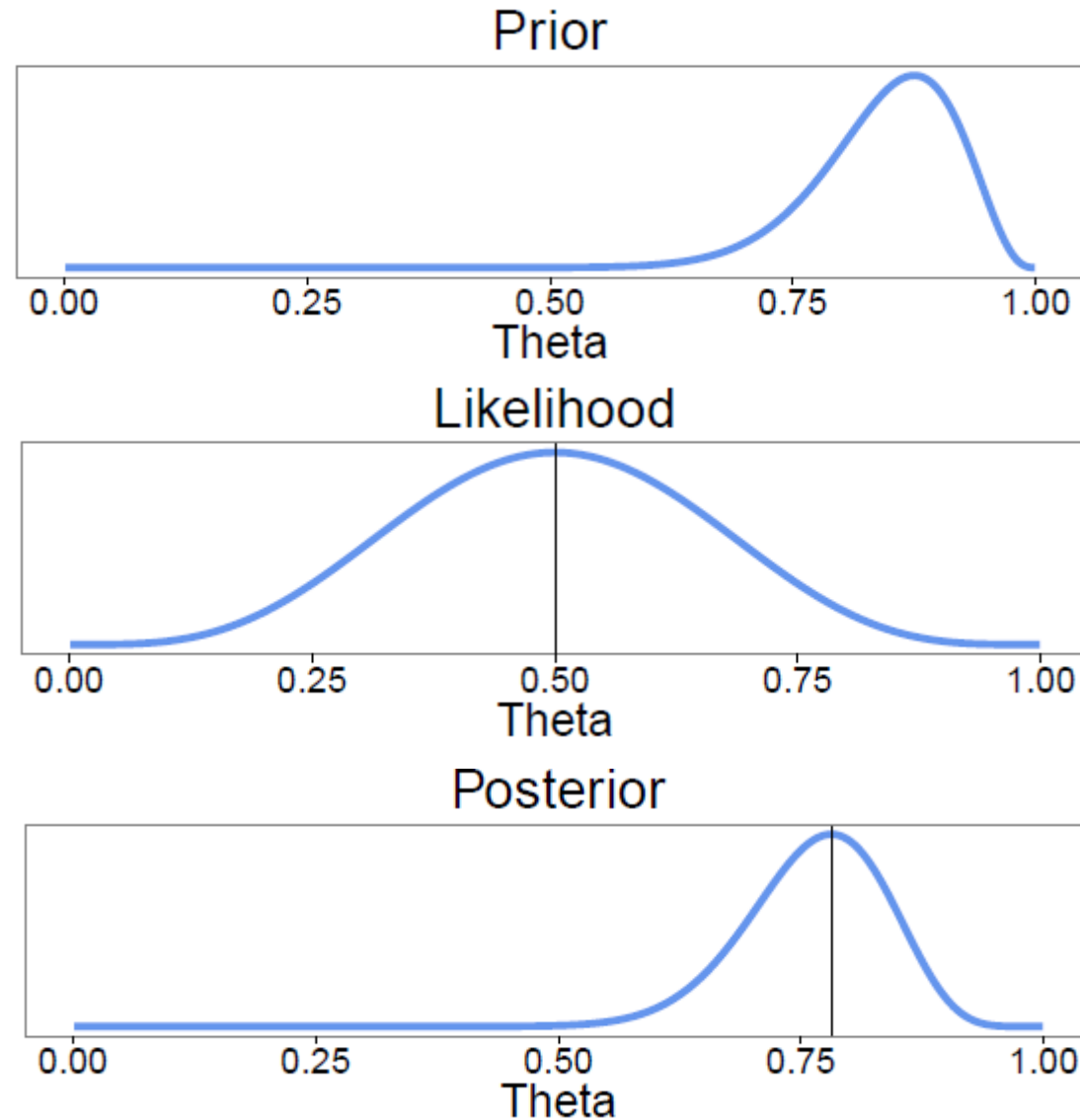
$p(\theta)$

Bernoulli likelihood function  
 $p(y|\theta)$

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

Figure 1. Results of simulating a random sample in a coin toss experiment using a Bernoulli likelihood under a true  $\theta$  of 0.5 and a sample size of 128.

# Relationship between MLE and MAP



Beta distribution  
 $p(\theta)$

Bernoulli likelihood function  
 $p(y|\theta)$

Beta distribution  
$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

Figure 2. In a small sample size of  $n = 8$ , the posterior is heavily influenced by a prior that is far off the true  $\theta$  of 0.5 (prior density parameterised with  $\alpha = 22, \beta = 4$ ).



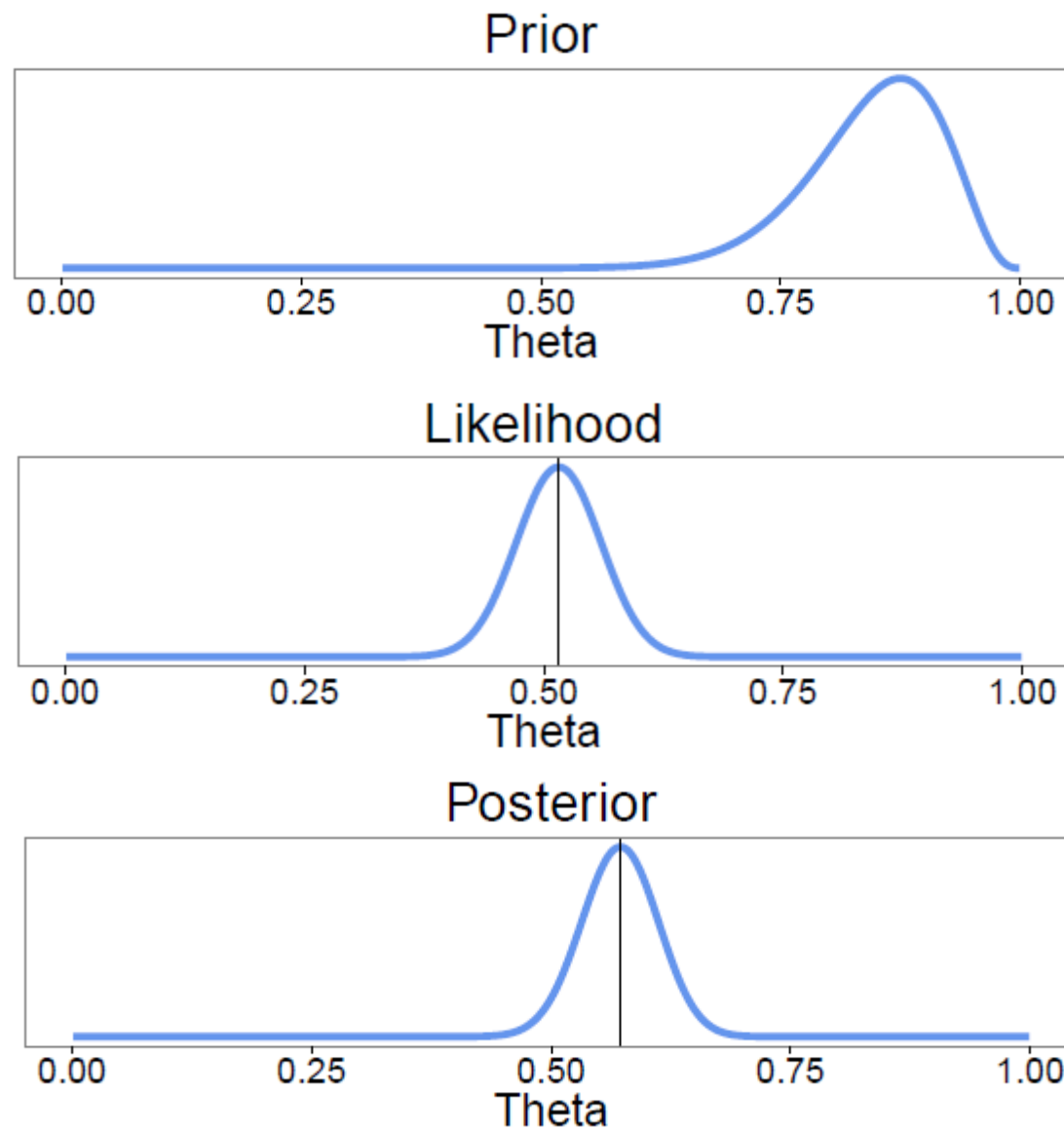
# Relationship between MLE and MAP

Thus, both estimators will become identical if sample size is large enough.

Formally: both estimators are **strong consistent**:

$$\hat{\theta}_{MLE} \rightarrow \theta [P]$$

$$\hat{\theta}_{MAP} \rightarrow \theta [P]$$



Beta distribution  
 $p(\theta)$

Bernoulli likelihood function  
 $p(y|\theta)$

Beta distribution

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

Figure 3. In a large sample size of  $n = 128$ , the posterior is much less influenced by a prior that is far off the true  $\theta$  of 0.5 (prior density parameterised with  $\alpha = 22, \beta = 4$ ).

# Some more terminology

Why use a beta distribution for the prior?

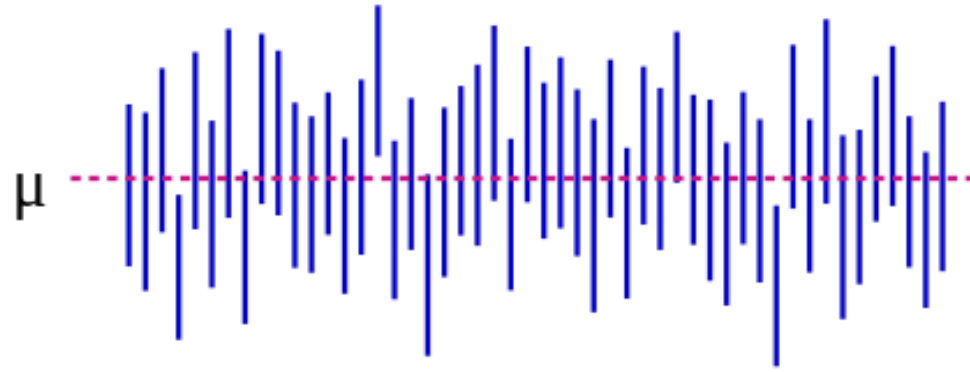
Beta distribution is a **conjugate prior** on a Bernoulli likelihood function.

- That means, if we choose a beta prior also our posterior will be a beta distribution.

If the prior and the posterior have the same distribution, the updating becomes very simple mathematically.

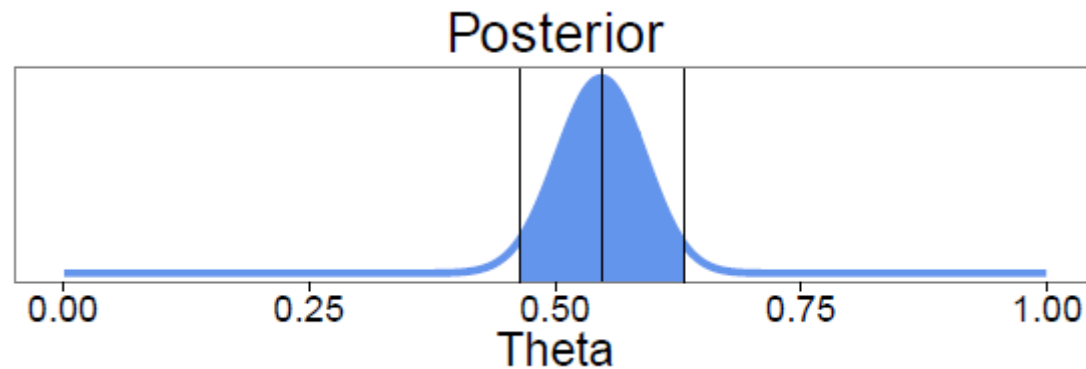
- There are conjugate priors for many of the most common likelihood functions.
- E.g., Gaussians are conjugate priors for Gaussian likelihoods, which is central for predictive coding models and the HGF

# Confidence intervals vs. highest density intervals



A proportion  $1 - \alpha$  (e.g. 95%) of the **confidence intervals** contain the true value.

(Note: bounds are random variables, parameter is unknown constant)



Smallest interval of a posterior distribution, which includes the mode of the distribution as well as a proportion of  $1 - \alpha$  of the mass, is called  $(1 - \alpha)$ -**credibility** or **highest density interval** (can be multimodal).

# Priors can be very helpful

Priors account for existing knowledge and prevent overfitting (being too close to the data)

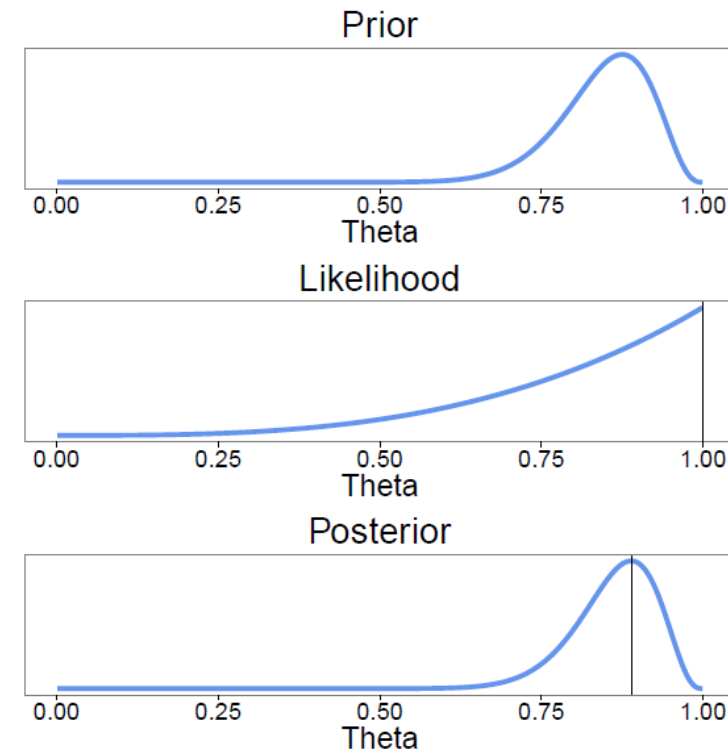
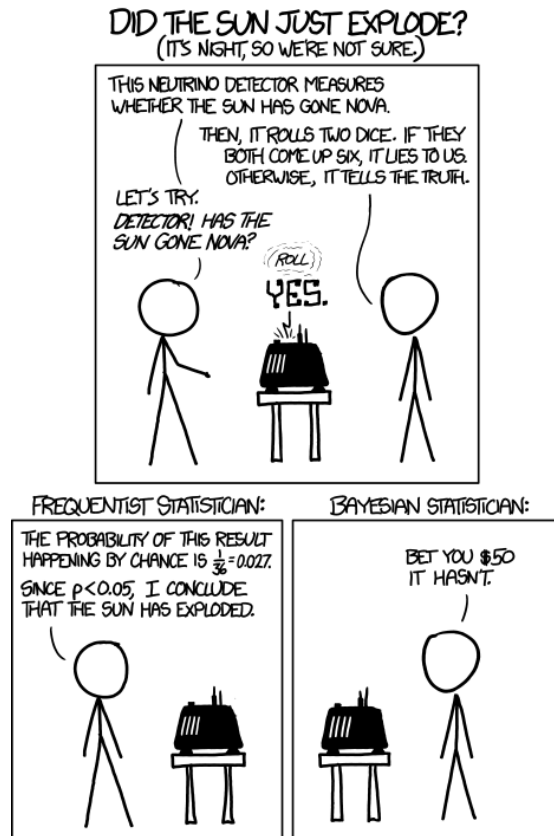


Figure 11. Even when we use a strongly biased prior, we find that the maximum a-posteriori estimate of  $\theta$  is much less biased by an extreme sample (3 heads in a row) than a maximum likelihood solution, which provides an estimate of  $\hat{\theta}_{MLE}(x) = 1$ . Note that this effect is particularly strong in small sample sizes. These problems can be connected to other statistical phenomena that have attracted much recent interest, such as the 'replication crisis' or the 'winner's curse' (e.g., Button et al., 2013)

# Priors can be very problematic

Usually, there are no formal rules of how to define a prior

- Particularly if parameter space is unbounded

What is the correct prior?

Some attempts:

- Jeffreys' prior (Jeffreys, 1946)
- Maximum entropy principle (Jaynes, 1968; 2003)

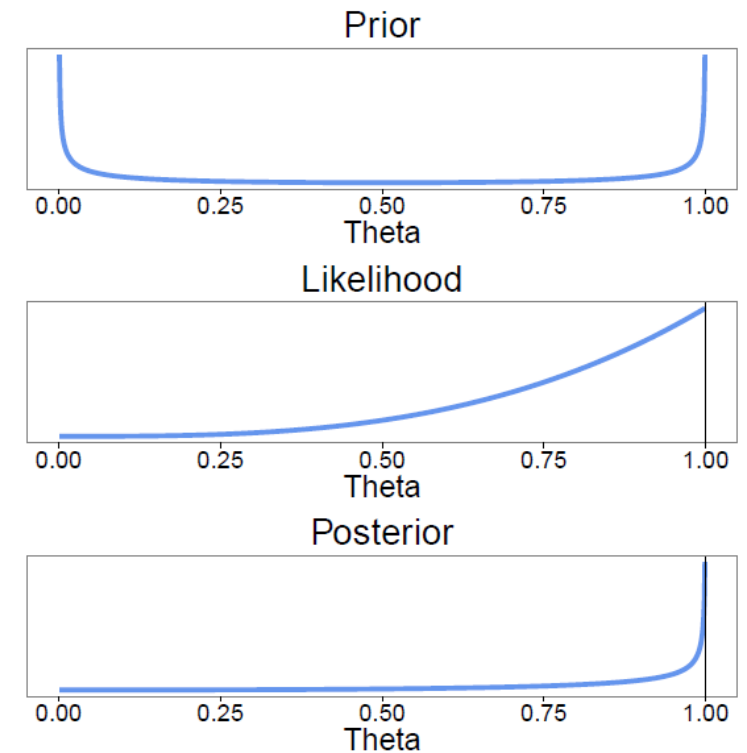


Figure 12. The Jeffreys prior for a coin toss experiment puts high probability mass on the extreme values of  $\Theta$ , thus providing a similar result as the maximum likelihood solution if the observe 3 successes in a row.

# Priors strongly affect what we can learn

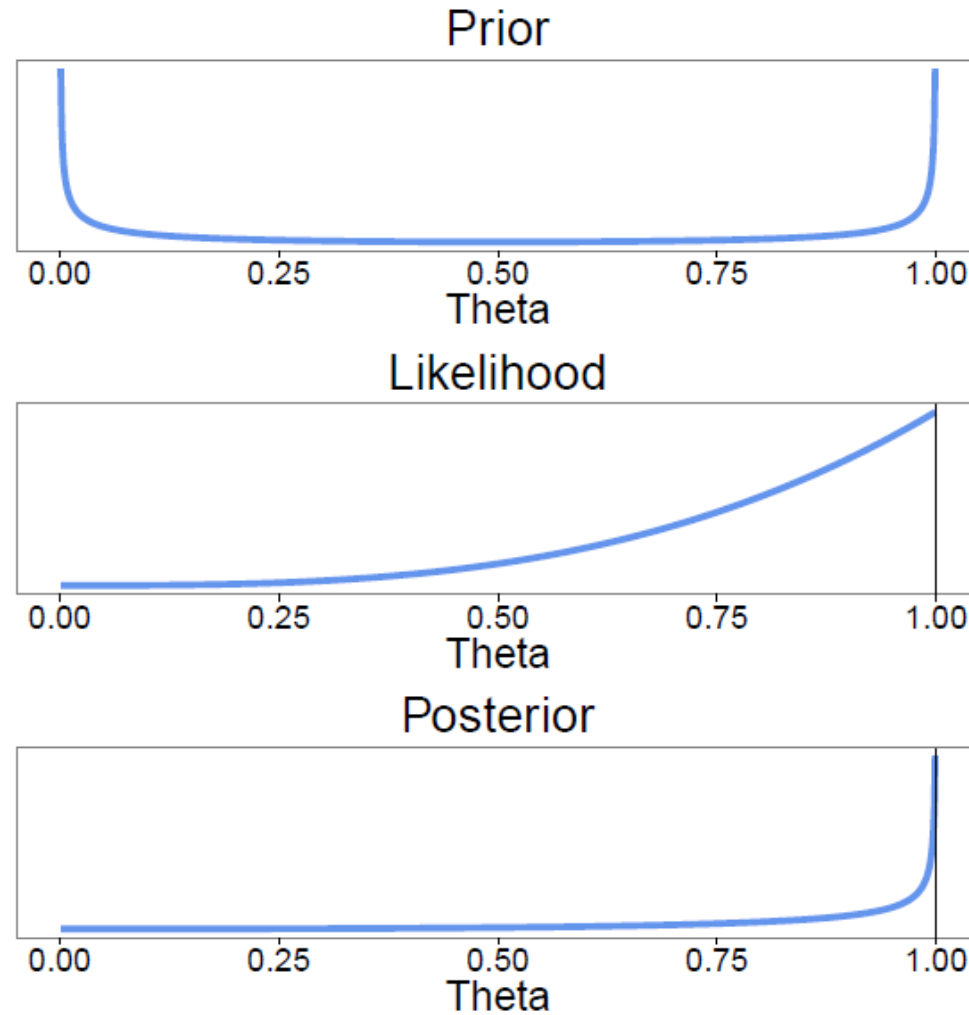


Figure 12. The Jeffreys prior for a coin toss experiment puts high probability mass on the extreme values of  $\Theta$ , thus providing a similar result as the maximum likelihood solution if the observe 3 successes in a row.

Break

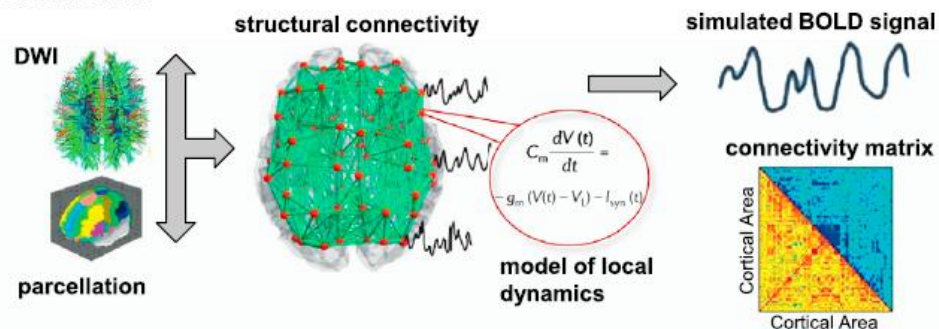
# Overview computational neuroscience

1. Develop a computational model of a cognitive/physiological process
  - Bayesian vs. non-Bayesian
2. Simulate data
3. Invert model on simulated data – estimate individual parameters
  - Bayesian vs. non-Bayesian
4. Optimise experimental design and collect data
5. Obtain individual parameters and possible group differences
6. Simulate behaviour of ‘computational quantities’
  - Use as regressors in imaging analysis

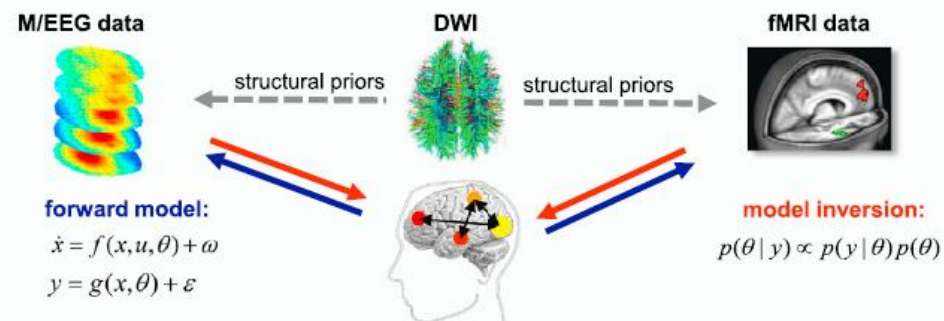


# Overview computational neuroscience

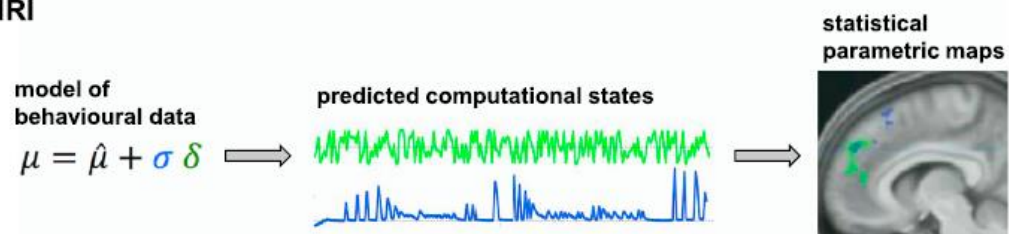
## A Biophysical network models



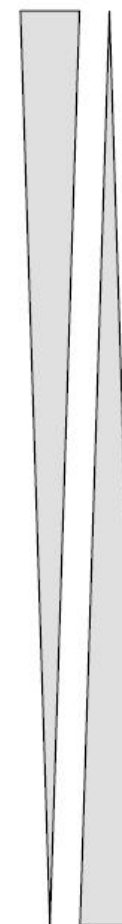
## B Generative Models



## C Model-based fMRI



Biological realism



Estimability

## 4. a) Worked Example: Context Inference.

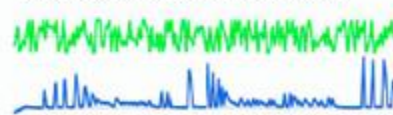
### C Model-based fMRI

model of  
behavioural data

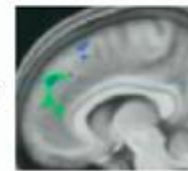
$$\mu = \hat{\mu} + \sigma \delta$$



predicted computational states



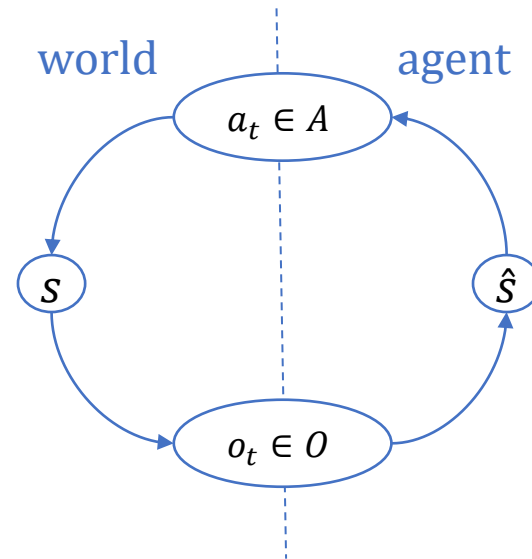
statistical  
parametric maps



# Building (generative) models of the world

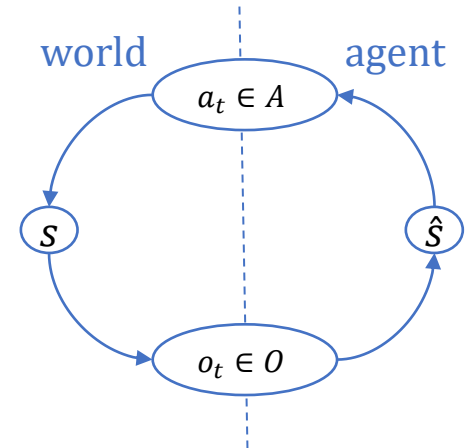
**Adaptive behaviour** requires agents to build probabilistic models of their environment  
(e.g., Conant & Ashby, 1970; Fiser et al., 2010; Friston, 2010)

- Represent statistical regularities
- **Bayesian brain hypothesis**



# (Somewhat) Interesting questions..

1. How do we develop these models in the first place?
  - i.e., how do we learn and generalise the structure of a problem (structure learning)?
2. How do we update representations?
  - Teaching signals, physiological implementation
3. How do we select the correct model of the world?
  - How do beliefs about the world translate into behaviour?
  - cf., Bayesian model comparison
4. Can this tell us something about pathological behaviour?
  - cf., Computational Psychiatry



[...]

# Building (generative) models of the world

How do we develop and update these probabilistic models?

- Specifically: What are candidate teaching signals that update our model?

Candidate I: Information-theoretic surprise

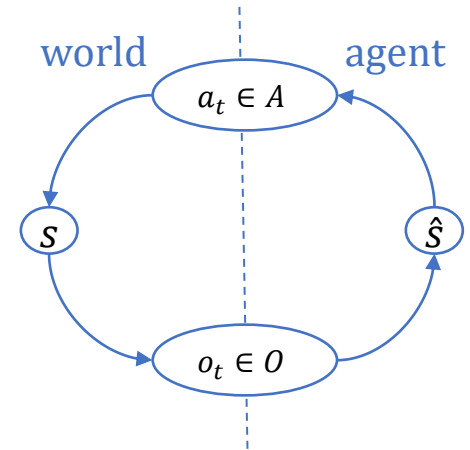
- Pure unexpectedness of a stimulus

$$-\ln p(o_t|m)$$

Candidate II: Actual (Bayesian) belief-updates

- Changes in model due to meaningful input

$$D_{KL} [p(s_t|o_t, m)||p(s_t|m)]$$



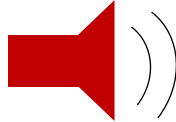
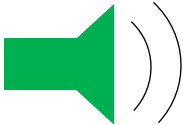


# Bayesian model averaging in behaviour and brain function?

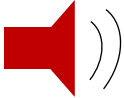
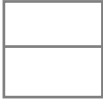

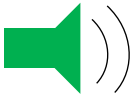
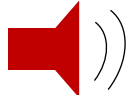

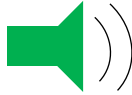


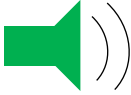
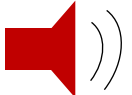

Simple decision paradigm:  
Accept or reject offer

Models  $\Leftrightarrow$  Contexts

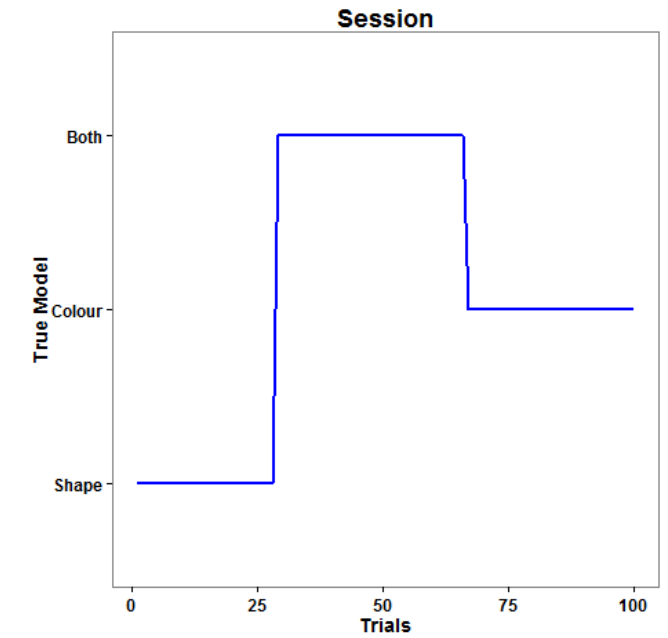
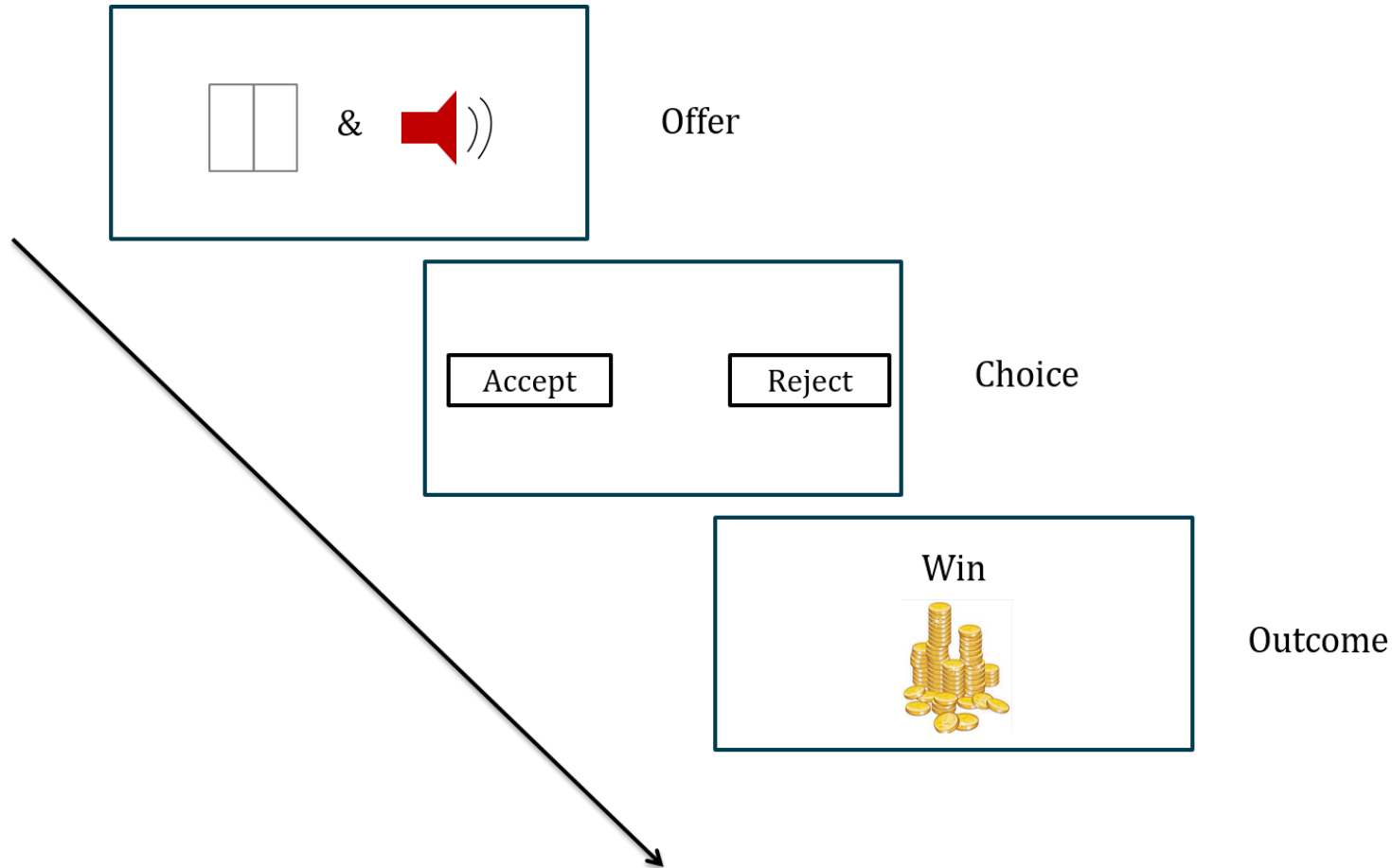
- Tone is important
- Shape is important
- Both is important

	Bad	Good
Shape		
Tone		

# Task: Context

P(win)	Tone	Shape	Both	
15%			<div> + </div> <div>[or]</div> <div> + </div>	Incongruent
85%			<div> + </div> <div>[or]</div> <div> + </div>	Congruent

# Task

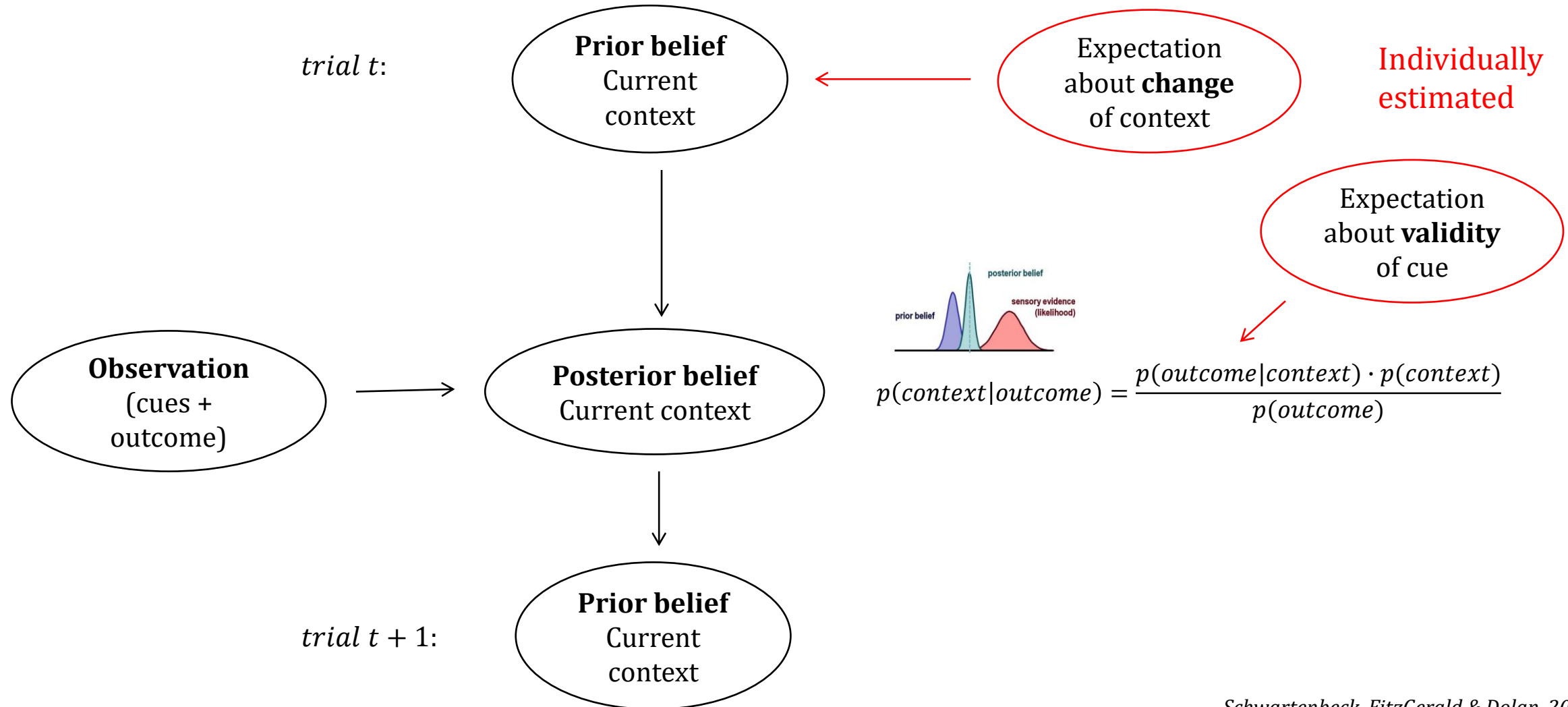


**Perform inference on:**

- I. Current context**
- II. Choice**

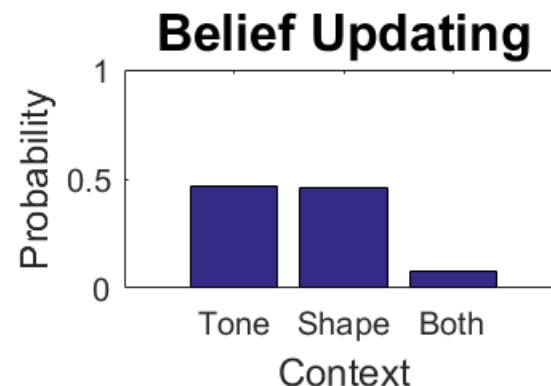
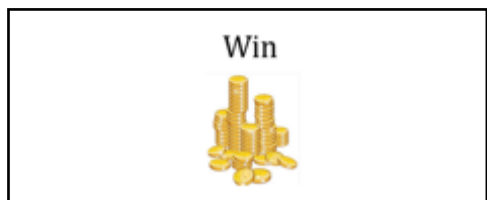
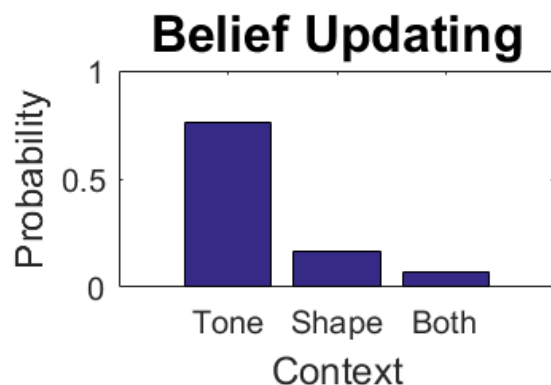
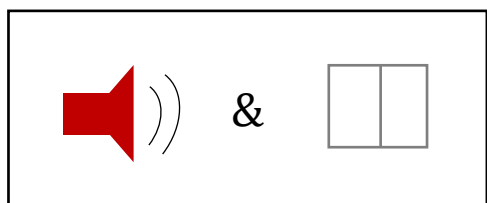


# Behavioural modelling



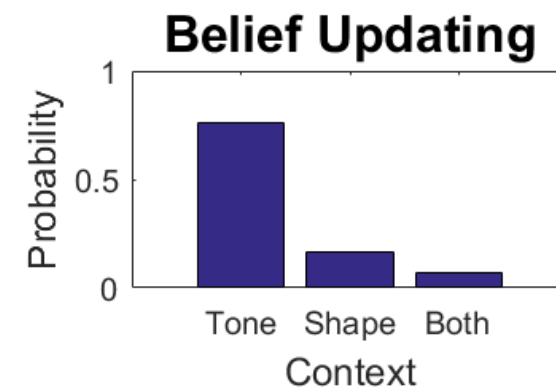
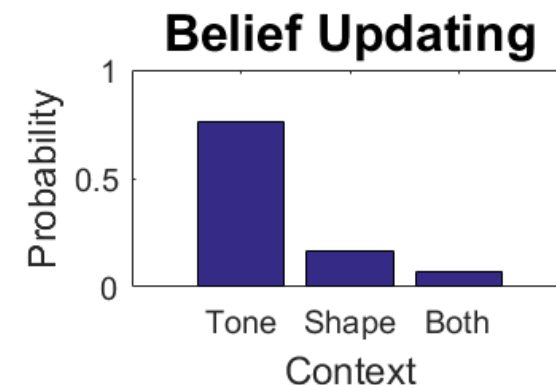
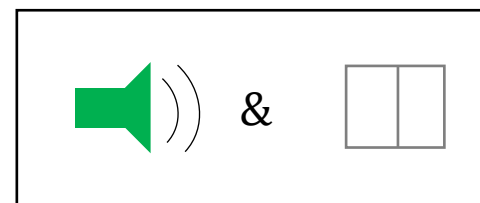
# Dissociating surprise from belief updates

## A) Informative trials:



↑ Surprise  
↑ Belief update

## B) Uninformative trials:



↑ Surprise

# Hands on Modelling!

Implement Bayesian belief updating within the for-loop of the ‘Experiment.m’ script

$$p(\theta|y) = \frac{p(y|\theta) \cdot p(\theta)}{p(y)}$$

$\theta \Leftrightarrow$  context

$y \Leftrightarrow$  observations (cue+outcome)

First, set up a prior for every trial  $i$

- Determine that at the first trial, the prior is uniform, i.e.  $p(\theta) = [\frac{1}{3} \frac{1}{3} \frac{1}{3}]$
- For any subsequent trial, determine that the prior of trial  $i$  is the posterior of trial  $i - 1$

# Hands on Modelling!

Implement Bayesian belief updating within the for-loop of the ‘experiment’ script

$$p(\theta|y) = \frac{p(y|\theta) \cdot p(\theta)}{p(y)} \quad \begin{array}{l} \theta \Leftrightarrow \text{context} \\ y \Leftrightarrow \text{observations (cue+outcome)} \end{array}$$

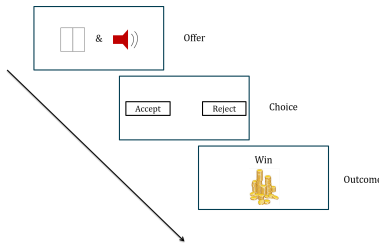
Next, determine the likelihood function for every trial  $i$

The likelihood should reflect the probability of observing a particular cue-outcome pair under every context:

$$p(y|\theta) = [p(\text{cue} + \text{outcome}|\text{tone}) \quad p(\text{cue} + \text{outcome}|\text{shape}) \quad p(\text{cue} + \text{outcome}|\text{both})]$$

- This depends on the validity of the cue!
- E.g., observing a bad tone and a good shape followed by a win should give a likelihood of:

$$p(y|\theta) = [0.15 \ 0.85 \ 0.15]$$



# Hands on Modelling!

Implement Bayesian belief updating within the for-loop of the ‘experiment’ script

$$p(\theta|y) = \frac{p(y|\theta) \cdot p(\theta)}{p(y)} \quad \begin{array}{l} \theta \Leftrightarrow \text{context} \\ y \Leftrightarrow \text{observations (cue+outcome)} \end{array}$$

Finally, compute the posterior of trial  $i$  based on the product of the likelihood and prior divided by the marginal likelihood  $p(y)$

- Note that  $p(y) = \sum_{\theta} p(y|\theta) \cdot p(\theta)$

Also, check how belief updates look like. How realistic do the changes in beliefs look like?

# Hands on Modelling!

One problem with the current model is that it is not aware that it has to do context updates.

- That means that beliefs can converge to 1 if we are in a context long enough (then updating is no longer possible)
- This also leads to really strange parameter estimates

We have to tell the model that the context can change

- One way to do this is to always reduce the current belief about the most likely context and increase the beliefs about being in one of the other two

Define a  $3 \times 3$  matrix (because there are 3 contexts)

- The diagonal elements should contain the probability of no reversal, i.e.  $1 - 3/60$  (because in 3 of 60 trials the context changes)
- The other elements should be the remaining probability  $(3/60)/2$
- Multiply the prior with this matrix at every trial  $i$  and check the belief updating again

# Hands on Modelling!

Now, define the trial-by-trial Kullback-Leibler divergence and (information theoretic) surprise at every trial.

The Kullback-Leibler divergence defines the actual changes in beliefs and is defined as

$$D_{KL} (posterior||prior) = \sum posterior \cdot \ln(\frac{posterior}{prior})$$

(but you can simply use the function 'KLD\_discrete')

The (information theoretic) surprise is simply defined as the negative log-likelihood of observing a cue-outcome pair under the current prior beliefs

- Note that this is simply the negative log-likelihood of the marginal likelihood in this case

# Hands on Modelling!

To make predictions about behaviour, we need to specify a mechanism of how beliefs about the task structure translate into behaviour.

One idea is Bayesian model averaging:

$$p(action) \approx \sum_{models} p(action|model) \cdot p(model|data)$$

Try to implement Bayesian model averaging in the code.

- That simply means multiplying the choice rule as implemented in the ‘rule’ matrix with the prior of trial  $i$



# Hands on Modelling!

We now have a (very basic) cognitive model that even makes some predictions for behaviour!

However, we may want to estimate some individual parameters to describe the observed behaviour of subjects.

- Here: *reversal probability* and *cue validity*

To do that, we need to make a proper function out of the ‘Experiment’ code.

- Get rid of all simulated events, and make the cognitive model depend on input that comes from calling the function ‘Experiment’

Finally, use the ‘ParameterEstimation.m’ script to estimate the parameters of some subjects

- Check what happens in the likelihood computation (end of ‘Estimation’ script)

# Hands on Modelling!

Now, explore the model and estimation (and try to find interesting ways of how to break the model)

For example:

- Effects of different parameters on belief updating
- MLE and MAP estimation (for map estimation uncomment the priors on the estimation section at the end of the 'Experiment' script)
- Effect of prior on MAP estimation (e.g. in subject 02, session 03)

You can also define different cognitive models and compare them

- AIC ( $= 2 \cdot \#parameters - 2 \cdot nll$ )
- BIC ( $= \ln n \cdot \#parameters - 2 \cdot nll$ )
- free energy...

# Overview computational neuroscience

1. Develop a computational model of a cognitive/physiological process
  - Bayesian vs. non-Bayesian
2. Simulate data
3. Invert model on simulated data – estimate individual parameters
  - Bayesian vs. non-Bayesian
4. Optimise experimental design and collect data
5. Obtain individual parameters and possible group differences
6. Simulate behaviour of ‘computational quantities’
  - Use as regressors in imaging analysis

## 4. b) Worked Example: Computational Psychiatry

# Computational Psychiatry

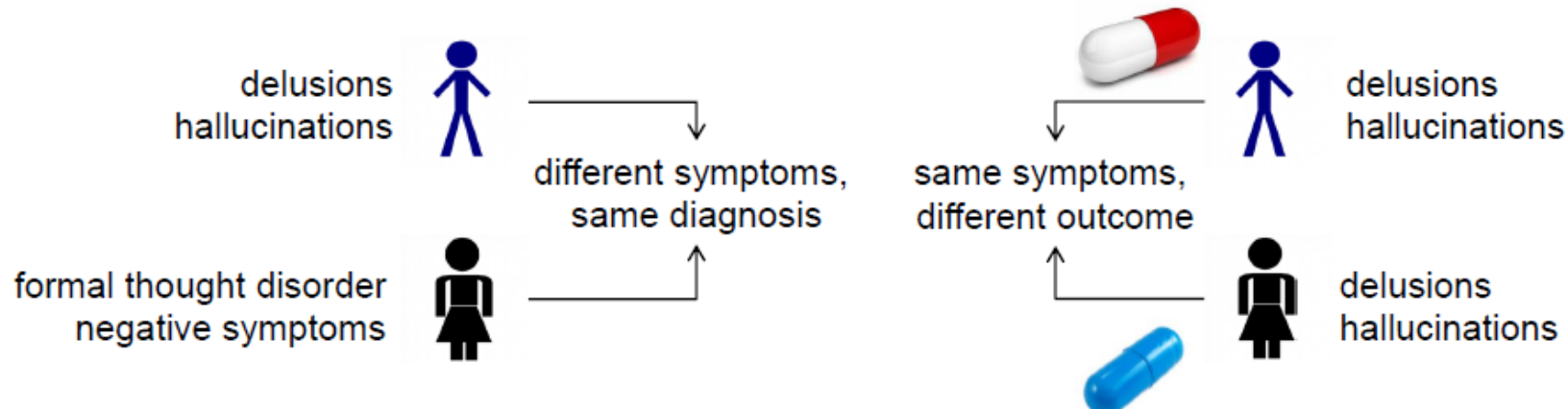
## DSM-IV: Schizophrenia

- Delusions
- Hallucinations
- Formal thought disorder
- Grossly disorganized or catatonic behavior
- Negative symptoms: flat affect, anhedonia, avolition, alogia, asociality

$\geq 2$  symptoms  
over  $\geq 1$  month

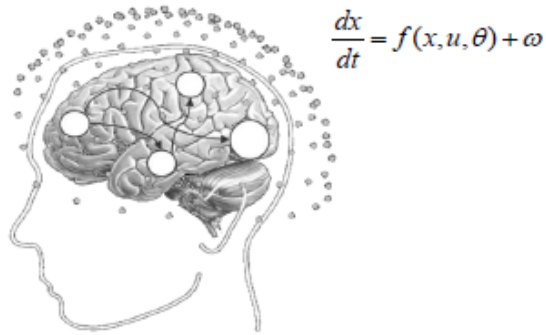
+ social or occupational dysfunction

+ continuous signs of the disturbance persist for at least six months



# Computational Psychiatry

- ❶ Computational assays:  
Models of disease mechanisms



- ❷ Application to brain activity and  
behaviour of individual patients



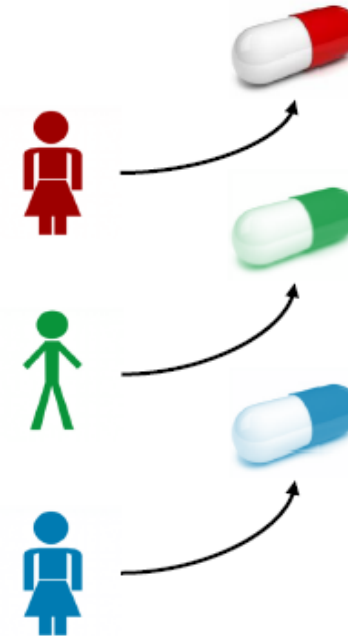
## Translational Neuromodeling

- ❸ Detecting physiological subgroups  
(based on inferred mechanisms)



- disease mechanism A
- disease mechanism B
- disease mechanism C

- ❹ Individual treatment prediction



# Computational Psychiatry

What can (Bayesian) computational models tell us about pathological behaviour and psychiatry?

Currently, diagnostic categorisations are based on phenomenological similarity

Computational models allow us to distinguish between different conditions

- Conditions that are fundamentally different despite superficial similarity

E.g., different ways to break:

- Broken inference
- Suboptimal model upon which inference is based

# Example computational psychiatry: Active Inference



## *Belief-based* (choice) behaviour

- Agents build **generative models** of their environment,
  - I.e. joint probability distributions over observations  $y$  and causes  $\theta$ :  $P(y, \theta) = P(y|\theta) \cdot P(\theta)$
- Agents invert these models to infer latent variables, including current states, policies and the precision of beliefs

## Objective function (cf., *computational level*, Marr): **minimise surprise**

- That means fulfilling expectations and maximising model evidence



# Subjective vs. objective generative models

Generative models are central for both modelling cognitive/neuronal responses as well as for data analysis

- $p(\theta, y) = p(y|\theta) \cdot p(\theta)$

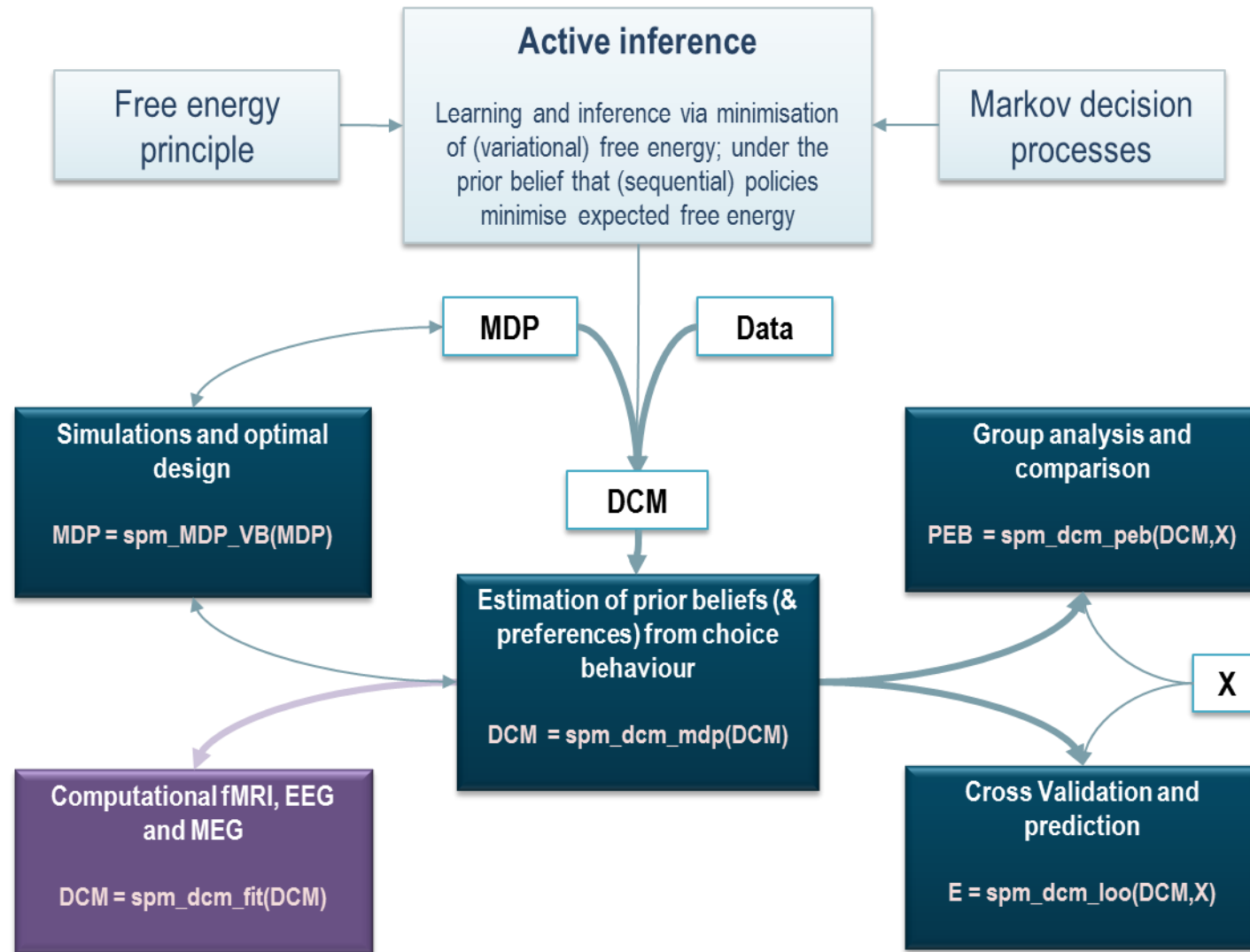
**Subjective generative model:** the model used by an agent to solve a task and perform inference

- Cognitive and/or neuronal
- Relevant data: task/problem

**Objective generative model:** analysis of observed behaviour or brain responses

- Relevant data: behaviour/neural data of subject in experiment

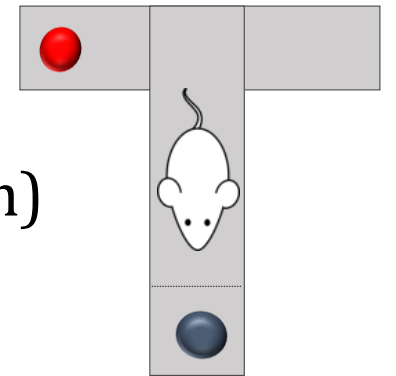
# Computational Phenotyping: Overview



# Computational Phenotyping: Task

## Two-step maze task

- A rat needs to obtain a reward in the left or right arm of a T-shaped maze
- It starts in the middle and can decide to go either left or right – or sample a cue at the bottom first
- The left and right arm are *absorbing states* (the rat cannot sample both)



Thus, the rat needs to solve the exploitation-exploration dilemma

- In case of high uncertainty, it should sample the cue first

# 1. Develop a computational model of a cognitive/physiological process

```

%% set up and preliminaries: first generate synthetic (single subject) data
=====
rng('default')

% outcome probabilities: A
%-----
% We start by specifying the probabilistic mapping from hidden states
% to outcomes.
%-----
a      = .95;
b      = 1 - a;
A      = [1 1 0 0 0 0 0 0;      % ambiguous starting position (centre)
          0 0 a b 0 0 0 0;      % left arm selected and rewarded
          0 0 b a 0 0 0 0;      % left arm selected and not rewarded
          0 0 0 0 b a 0 0;      % right arm selected and rewarded
          0 0 0 0 a b 0 0;      % right arm selected and not rewarded
          0 0 0 0 0 0 1 0;      % informative cue - reward on right
          0 0 0 0 0 0 0 1];    % informative cue - reward on left

% controlled transitions: B{u}
%-----
% Next, we have to specify the probabilistic transitions of hidden states
% under each action or control state. Here, there are four actions taking the
% agent directly to each of the four locations.
%-----
B{1} = [1 0 0 1; 0 1 0 0; 0 0 1 0; 0 0 0 0]; % move to the middle
B{2} = [0 0 0 0; 1 1 0 1; 0 0 1 0; 0 0 0 0]; % move up left (and check for reward)
B{3} = [0 0 0 0; 0 1 0 0; 1 0 1 1; 0 0 0 0]; % move up right (and check for reward)
B{4} = [0 0 0 0; 0 1 0 0; 0 0 1 0; 1 0 0 1]; % move down (check cue)

for i = 1:4
    B{i} = kron(B{i}, eye(2));
end

```

```

% priors: (utility) C
%-----
% Finally, we have to specify the prior preferences in terms of log
% probabilities. Here, the agent prefers rewarding outcomes
%-----
c      = 2;
C      = [0 c -c c -c 0 0]';

% now specify prior beliefs about initial state, in terms of counts
%-----
d      = kron([1 0 0 0], [1 1])';

% allowable policies (of depth T). These are just sequences of actions
%-----
V      = [1 1 1 1 2 3 4 4 4 4
          1 2 3 4 2 3 1 2 3 4];

```

```

%% MDP Structure - this will be used to generate arrays for multiple trials
=====
mdp.V = V; % allowable policies
mdp.A = A; % observation model
mdp.B = B; % transition probabilities
mdp.C = C; % preferred states
mdp.D = d; % prior over initial states
mdp.s = 1; % initial state

mdp.alpha = 2; % precision of action selection
% mdp.alpha = 8; % precision of action selection
mdp.beta = 1; % inverse precision of policy selection

% true parameters
%-----
n      = 128; % number of trials
i      = rand(1,n) > 1/2; % randomise hidden states over trials
P.beta = log(2);
P.C     = log(2);

MDP     = mdp;
% MDP.C = mdp.C;
MDP.C   = mdp.C*exp(P.C);
MDP.beta = mdp.beta*exp(P.beta);

[MDP(1:n)] = deal(MDP);
[MDP(i).s] = deal(2);

```

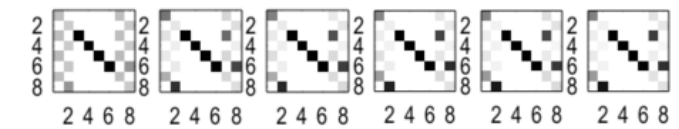
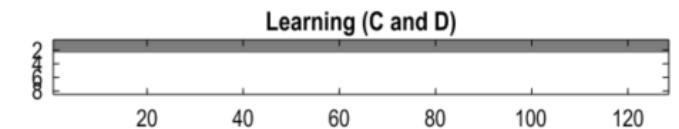
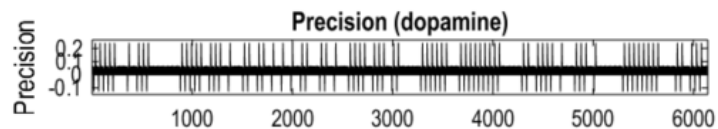
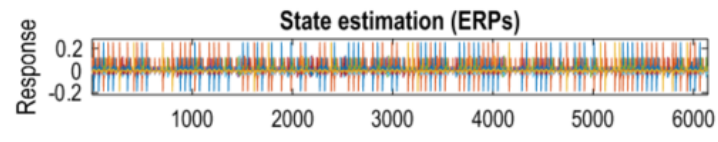
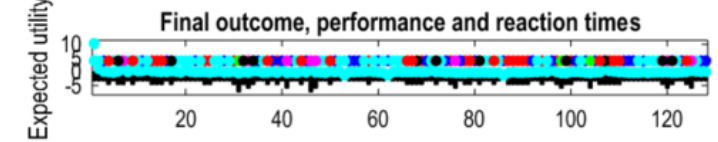
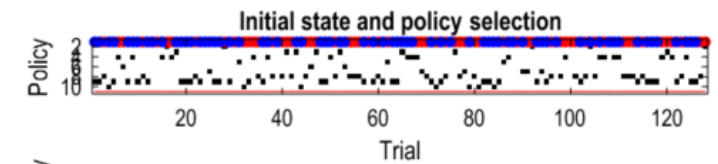
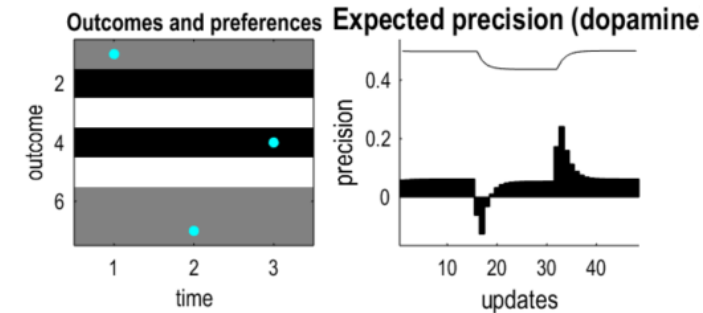
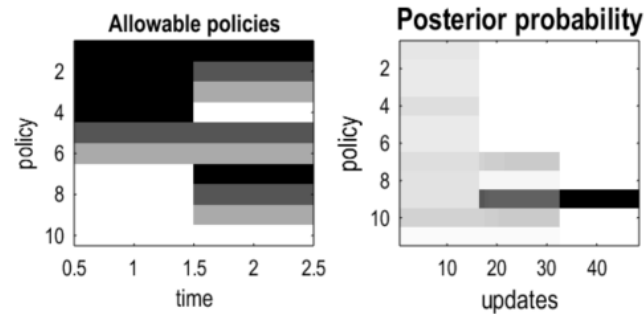
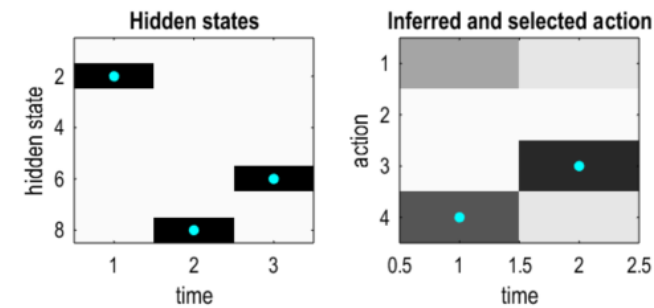
## 2. Simulate data

```
%% Solve to generate data
%=====
MDP = spm_MDP_VB(MDP);

% illustrate behavioural responses - single trial
%-----
spm_figure('GetWin','Figure 1a'); clf
spm_MDP_VB_trial(MDP(1));

% illustrate behavioural responses and neuronal correlates over trials
%-----
spm_figure('GetWin','Figure 1b'); clf
spm_MDP_VB_game(MDP);

%-----
% This completes the generation of data. We now turn to the estimation of
% subject specific preferences and precision encoded by the parameters
% beta and C. Model parameters here are log scaling parameters that allow
% for increases or decreases in the default prior values.
%-----
```

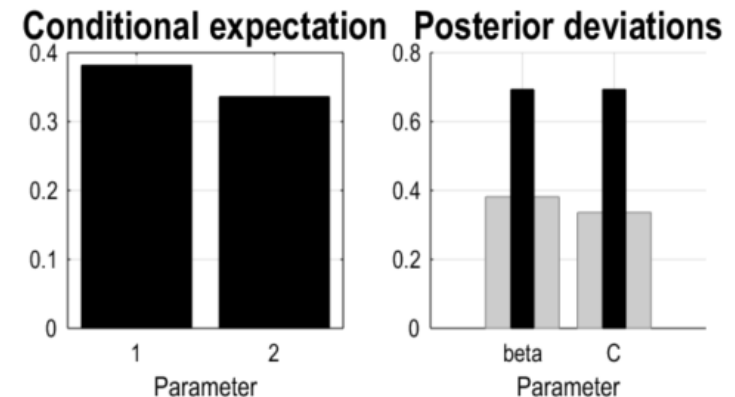
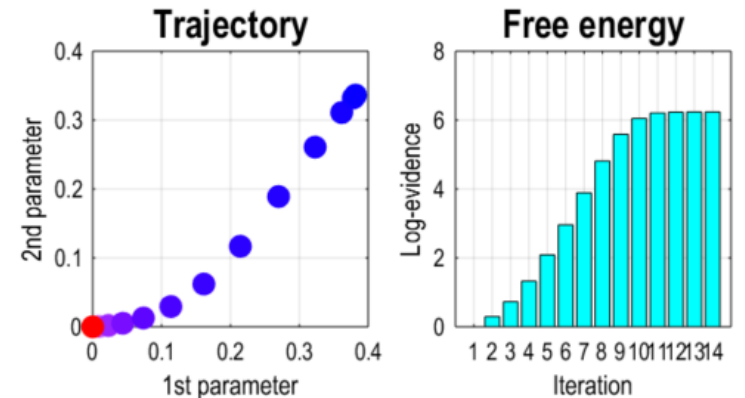


# 3. Invert model on simulated data – estimate individual parameters

```
%% Invert to recover parameters (preferences and precision)
%=====
DCM.MDP = mdp; % MDP model
DCM.field = {'beta', 'C'}; % parameter (field) names to optimise
DCM.U = {MDP.o}; % trial specification (stimuli)
DCM.Y = {MDP.u}; % responses (action)

DCM = spm_dcm_mdp(DCM);

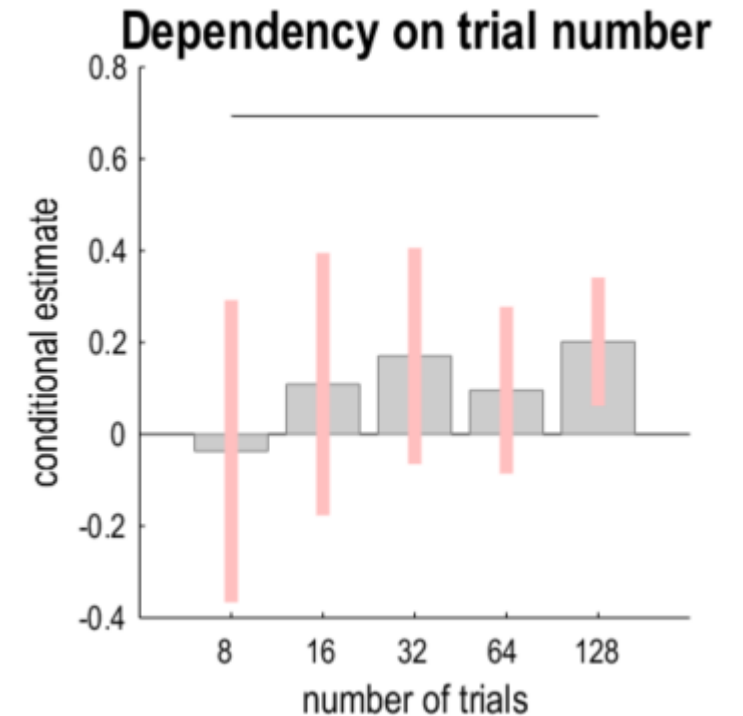
% compare true values with posterior estimates
%-----
subplot(2,2,4), hold on
bar(spm_vec(P), 1/4)
set(gca, 'XTickLabel', DCM.field)
set(gcf, 'Name', 'Figure 2', 'Tag', 'Figure 2')
```



# 4. Optimise design parameters (sample size)

```
%% now repeat using subsets of trials to illustrate effects on estimators - design optimisation!
%=====
DCM.field = {'beta'};
n         = [8 16 32 64 128];
for i = 1:length(n)
    DCM.U = {MDP(1:n(i)).o};
    DCM.Y = {MDP(1:n(i)).u};
    DCM   = spm_dcm_mdp(DCM);
    Ep(i,1) = DCM.Ep.beta;
    Cp(i,1) = DCM.Cp;
end

% plus results
%-----
spm_figure('GetWin','Figure 3'); clf
subplot(2,1,1), spm_plot_ci(Ep(:),Cp(:)), hold on
plot(1:length(n),(n - n) + P.beta,'k'), hold off
set(gca,'XTickLabel',n)
xlabel('number of trials','FontSize',12)
ylabel('conditional estimate','FontSize',12)
title('Dependency on trial number','FontSize',16)
axis square
```



# 5. (Collect data and) Obtain individual parameters

```
% now repeat but over multiple subjects with different beta
%-----

% generate data and a between subject model with two groups of eight
% subjects
%-----
N = 8; % numbers of subjects per group
X = kron([1 1; 1 -1], ones(N,1)); % design matrix
h = 4; % between subject log precision
n = 128; % number of trials
i = rand(1,n) > 1/2; % randomise hidden states

clear MDP
[MDP(1:n)] = deal(mdp);
[MDP(i).s] = deal(2);

reward = zeros(n,size(X,1));

for i = 1:size(X,1)

    % true parameters - with a group difference of one half
    %-----
    beta(i) = X(i,:)*[0; 1/4] + exp(-h/2)*randn; % add random Gaussian effects to group means -> BMR and PEB
    % beta(i) = X(i,:)*[0; 0] + exp(-h/2)*randn; % add random Gaussian effects to group means -> BMR and PEB
    [MDP.beta] = deal(exp(beta(i)));

    % solve to generate data
    %-----
    DDP = spm_MDP_VB(MDP); % realisation for this subject
    DCM.U = {DDP.o}; % trial specification (stimuli)
    DCM.Y = {DDP.u}; % responses (action)
    GCM{i,1} = DCM;

    for kk=1:length(DCM.U)
        if DCM.U{kk}(end)==2 || DCM.U{kk}(end)==4 % outcome 2 or 4 == reward
            reward(kk,i)=1;
        end
    end

    % plot behavioural responses
    %-----
    spm_figure('GetWin','Figure 4'); clf
    spm_MDP_VB_game(DDP); drawnow

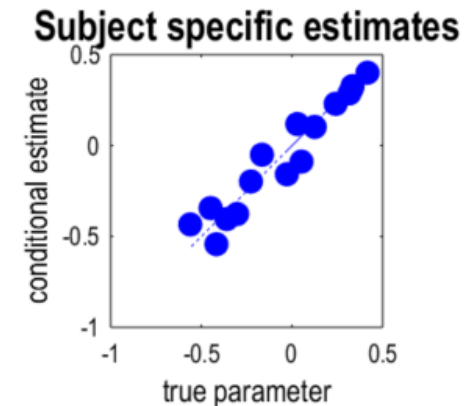
end
```

```
% Bayesian model inversion
%=====

GCM = spm_dcm_fit(GCM);

% plot subject specific estimates and true values
%-----

spm_figure('GetWin','Figure 4');
subplot(3,1,3)
for i = 1:length(GCM)
    qP(i) = GCM{i}.Ep.beta;
end
plot(beta,beta,':b',beta,qP,':b','MarkerSize',32)
xlabel('true parameter','FontSize',12)
ylabel('conditional estimate','FontSize',12)
title('Subject specific estimates','FontSize',16)
axis square
```



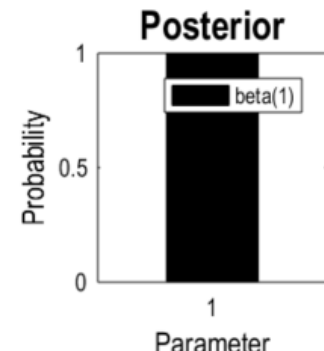
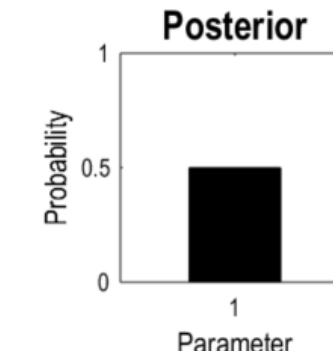
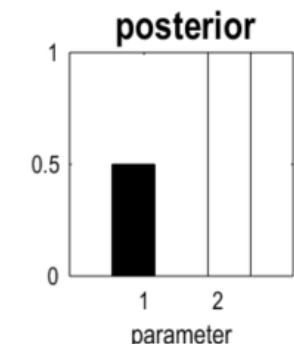
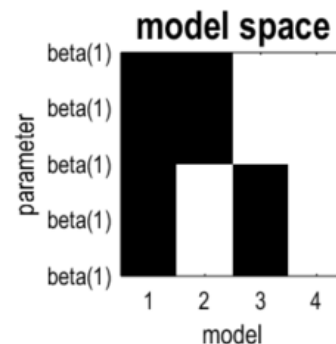
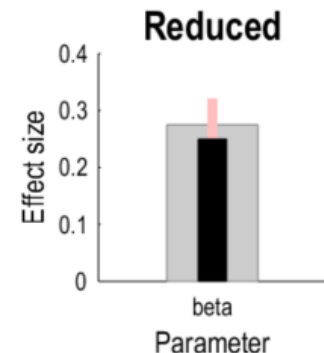
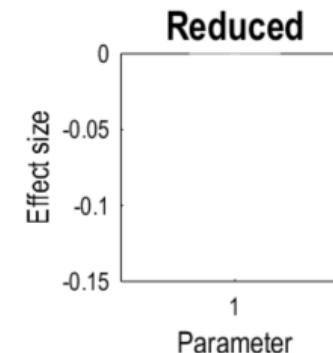
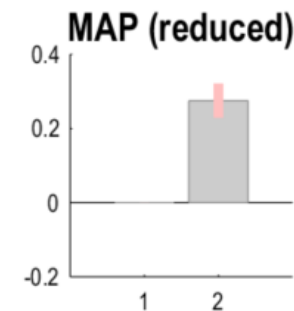
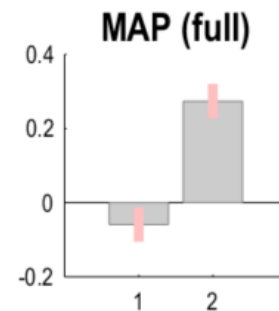
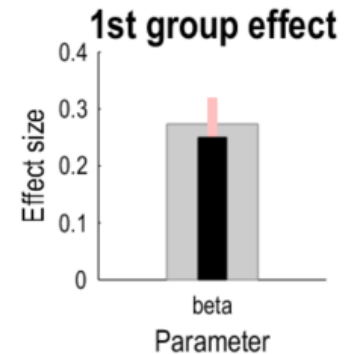
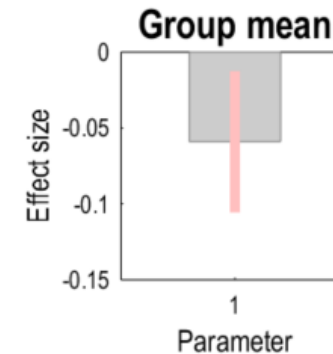
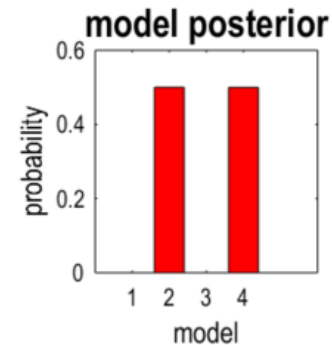
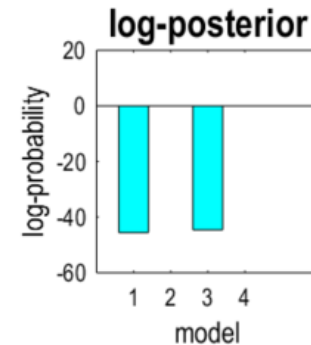


# 6. Estimate group differences in parameters

```

%% hierarchical (empirical) Bayes
=====
% second level model
%
M = struct('X',X);
%
% BMA - (second level)
%
PEB = spm_dcm_peb(GCM,M);
BMA = spm_dcm_peb_bmc(PEB);

subplot(3,2,4),hold on, bar(1,1/4,1/4), set(gca,'XTickLabel',DCM.field)
subplot(3,2,2),hold on, bar(1,1/4,1/4), set(gca,'XTickLabel',DCM.field)
    
```

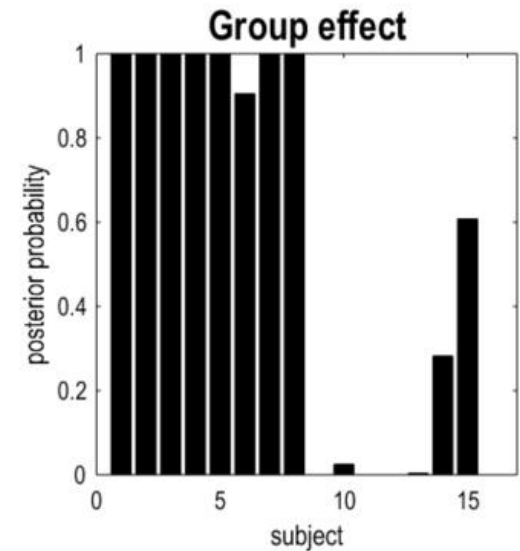
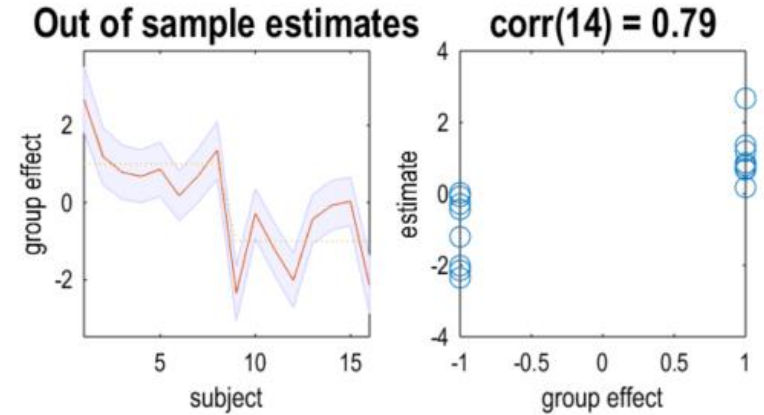


## 6. Estimate group differences in parameters

```
%% posterior predictive density and cross validation
```

```
%=====
```

```
spm_dcm_loo(GCM,M,DCM.field);
```



# Overview computational neuroscience

1. Develop a computational model of a cognitive/physiological process
  - Bayesian vs. non-Bayesian
2. Simulate data
3. Invert model on simulated data – estimate individual parameters
  - Bayesian vs. non-Bayesian
4. Optimise experimental design and collect data
5. Obtain individual parameters and possible group differences
6. Simulate behaviour of ‘computational quantities’
  - Use as regressors in imaging analysis

# Hype: Computational Modelling

Using modelling does not replace thoughts about experimental design

Modelling does not necessarily make the results more substantial

Modelling does not necessarily clarify things

- E.g., verbal inaccuracy vs. ambiguous role of parameters

But:

- Modelling forces you to think more deeply about processes and hypothesis

# Hype: Bayes

Bayes can make things very complicated

- Computationally intensive
- Not always clear how to specify prior

It's very hard - or perhaps impossible - to falsify

Does Bayes provide an explanatory account of cognition/brain function, or merely a description?

There really is nothing wrong with frequentist statistics and p-values

- As long as you know what they do (and cannot do)

# Hype: Computational Psychiatry

Will computational psychiatry ever translate into clinical practice?

Patients rely on us, but do we do enough for them?  
A (very) personal view on Computational Psychiatry

# Take home messages

The process of building models is quite central.

- Relevant for understanding (experimental) data
- Relevant for navigating successfully in the world
- Bayes can be helpful here

Bayes provides a framework for integrating information.

- Provides interesting models of cognition/brain function

Computational modelling can be useful for teasing apart otherwise unobservable processes.

When we try to infer the context in a gambling task, we probably rely on Bayesian inference.

# Many thanks to:



Karl Friston (UCL)  
Ray Dolan (UCL)

Martin Kronbichler (CCNS)

Thomas FitzGerald (UEA)  
Christoph Mathys (SISSA)



...and thank you for listening!





# Probability theory basics: Bayes' theorem

## Bayes' Theorem

Based on the concept of a probability density function, we can now introduce a more general form of Bayes' theorem (Felsenstein, 2008):

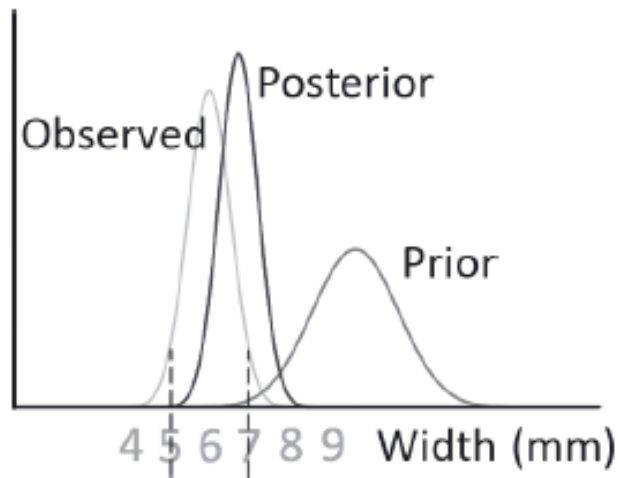
$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta}.$$

# What is a prior?

$$p(\theta)$$

A prior density reflects any existing a-priori knowledge about an unknown quantity.

A prior can be thought of as ‘regularising’ the likelihood:

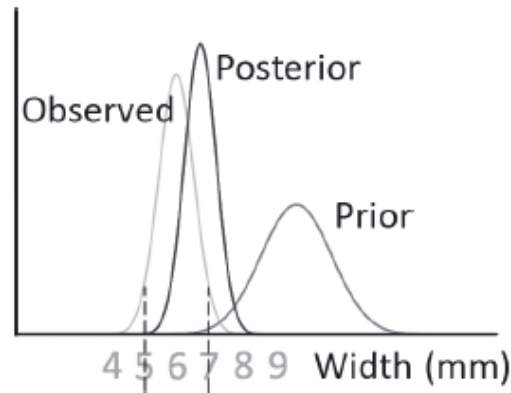


# What is the likelihood?

$$p(y|\theta)$$

The likelihood reflects the probability of observing the data under the current hypothesis.

Without any prior knowledge posterior = likelihood



Not a probability distribution!

# What is the marginal likelihood?

$p(y)$

Also called ‘evidence’

$\theta$  is marginalised out and thus reduces the denominator to

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta} \quad \int f(x|\theta) \cdot \pi(\theta)d\theta = f(x)$$

Serves as a normalisation factor to ensure that the posterior distribution  $\pi(\theta|x)$  sums to one

# Marginal likelihood : some terminology

$$\int f(x|\theta) \cdot \pi(\theta) d\theta = f(x)$$

Integral usually cannot be solved analytically and has to be approximated

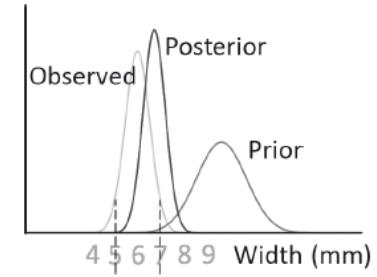
Popular approaches:

- **Markov chain Monte Carlo** (*e.g., Sanborn & Chater, 2016*)
- **Variational Bayes** (*e.g., Friston et al., 2007; Mathys et al., 2012*)

# What is the posterior?

$$p(\theta|y)$$

Update of the prior distribution after making an observation



Posterior density has on average smaller variance and entropy than the prior density (Felsenstein, 2008).

Because marginal likelihood is just a normalisation constant, Bayes' rule is also often written as

$$p(\theta|y) \propto p(y|\theta) \cdot p(\theta)$$

Computing the posterior can also be understood as inverting the model  $p(y|\theta) \cdot p(\theta)$

- i.e., estimate parameter  $\theta$

# Confidence intervals vs. highest density intervals

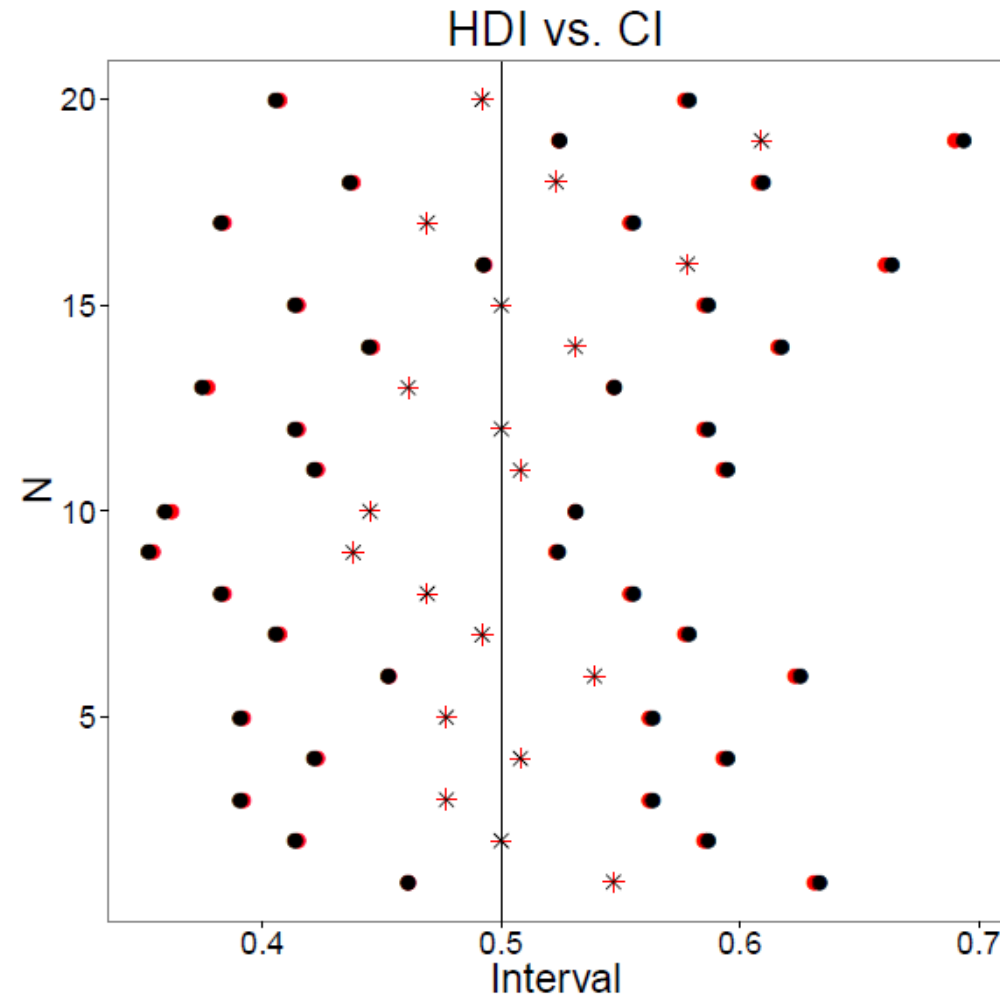
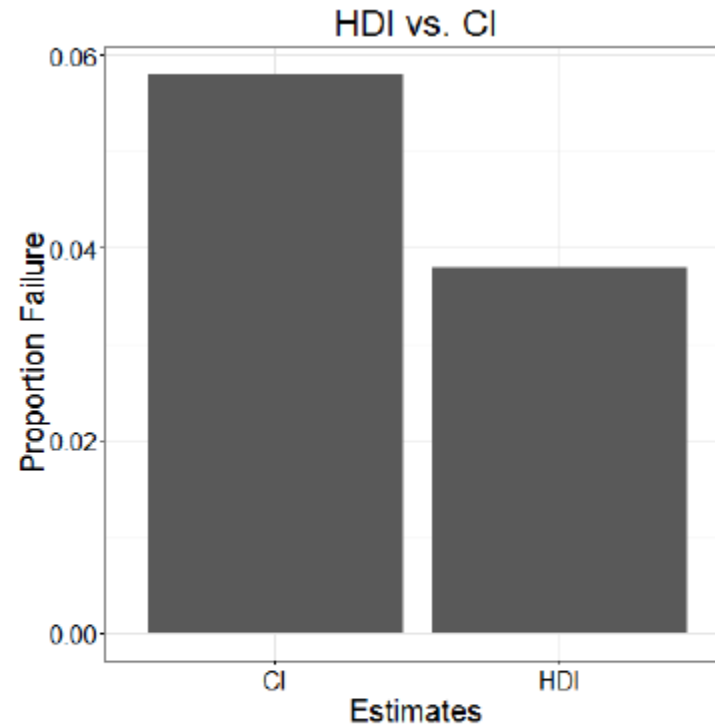


Figure 4. Comparing 20 95% confidence intervals (black dots) around ML estimates (black asterisks) and credible intervals (red dots) around MAP estimates (red crosses) under a uniform prior and a true  $\theta$  of 0.5 (black vertical line) in simulations with a sample size of  $n = 128$ .



# Confidence intervals vs. highest density intervals



*Figure 5. Comparing the proportion of failure to capture the true  $\theta$  of 0.5 in confidence intervals and credible intervals under a uniform prior (expected proportion of failure is 0.05, 1000 simulations using a sample size of 128). We see that both approaches have a very similar proportion of failure that is close to the expected value.*

# Confidence intervals vs. highest density intervals

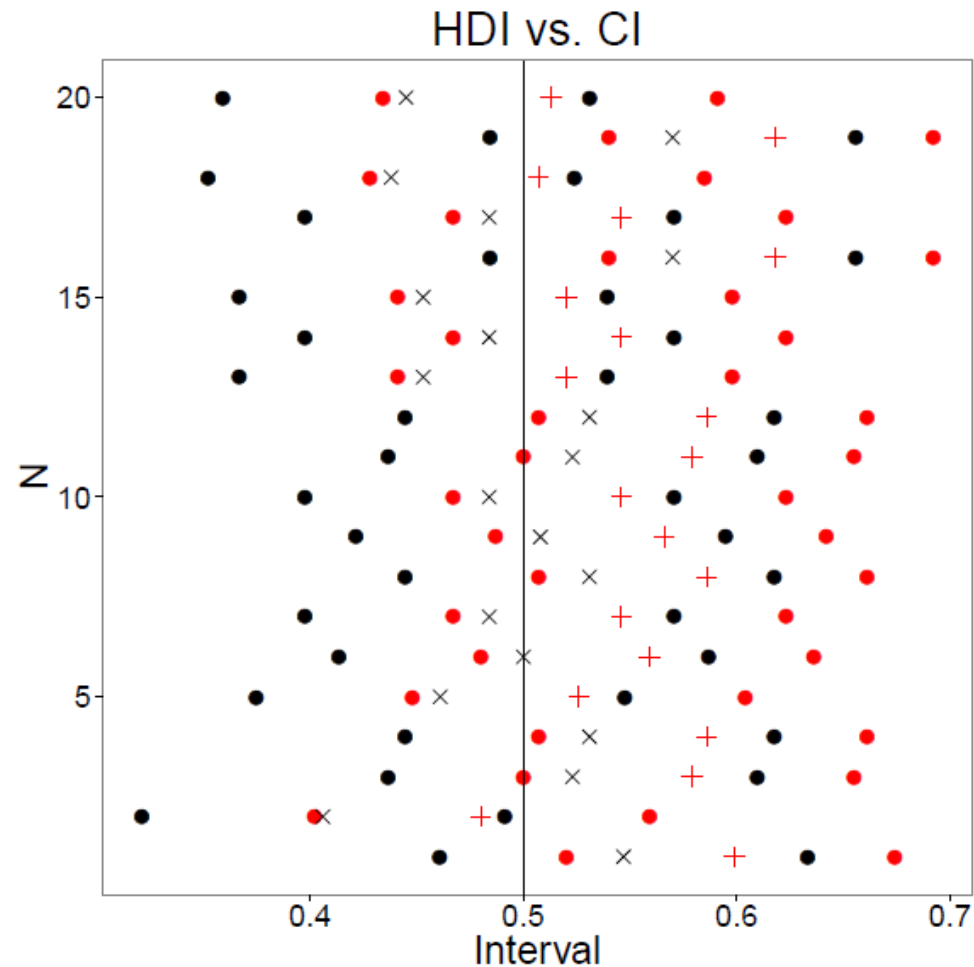


Figure 8. Comparing 95% confidence intervals (black dots) around ML estimates (black asterisks) and credible intervals (red dots) around MAP estimates (red crosses) under a prior that is strongly biased towards higher values of  $\theta$  as shown in Figures 2 and 3 but using a larger sample size of  $n = 128$  and a true  $\theta$  of 0.5 (black vertical line). We see that in larger sample sizes the posterior distribution is much less affected by 'malicious' priors, such that the credible capture the true parameter value much more often.

# Confidence intervals vs. highest density intervals

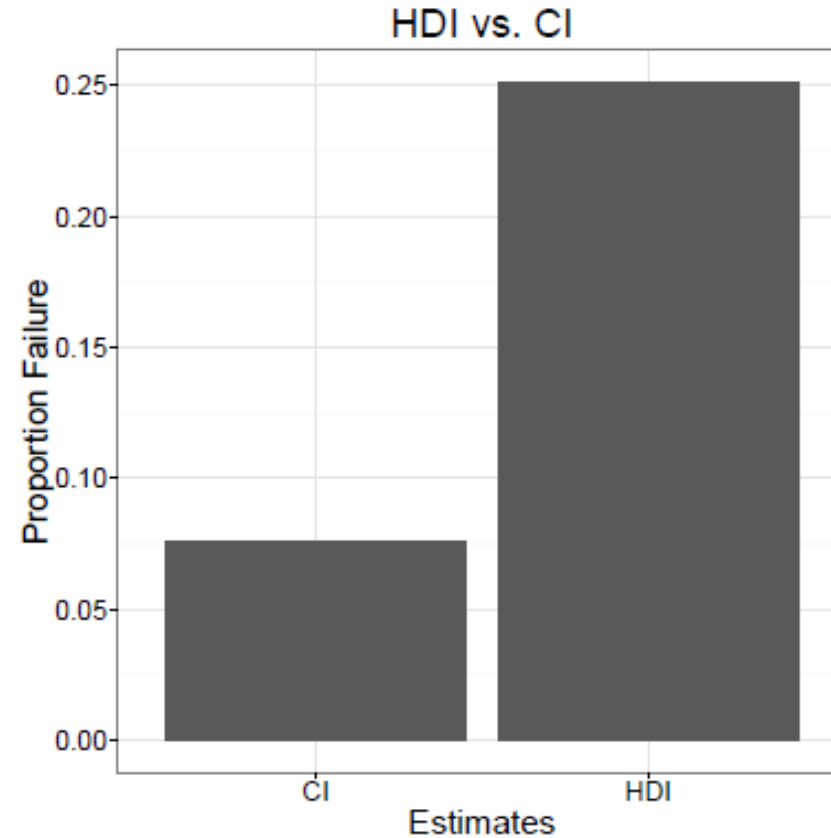
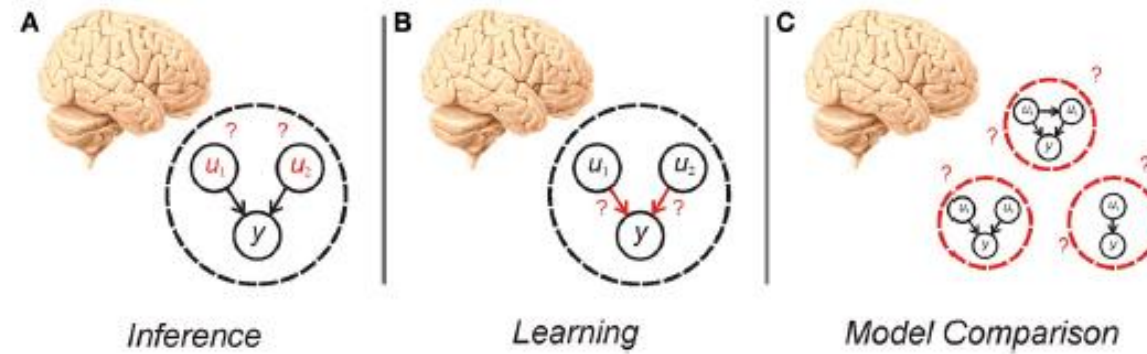


Figure 9. Comparing the proportion of failure to capture the true  $\theta$  of 0.5 for confidence intervals and credible intervals under a prior that is strongly biased towards higher values of  $\theta$  as shown in Figures 2 and 3 but using a large sample size of  $n = 128$  (expected proportion of failure is 0.05). We observe that the proportion of failure of Bayesian intervals is about four times higher than the proportion of failure of frequentist intervals.

# Building (generative) models of the world



We try to find accurate *and* simple models.