Hierarchical Bayesian modelling of cognition

Dr. Lei Zhang

Associate Professor
Adaptive **L**earning **P**sychology & **N**euroscience (ALPN) Lab
Centre for Human Brain Health, School of Psychology
University of Birmingham

lei-zhang.net
@lei_zhang_lz

SZU, 20.11.2025

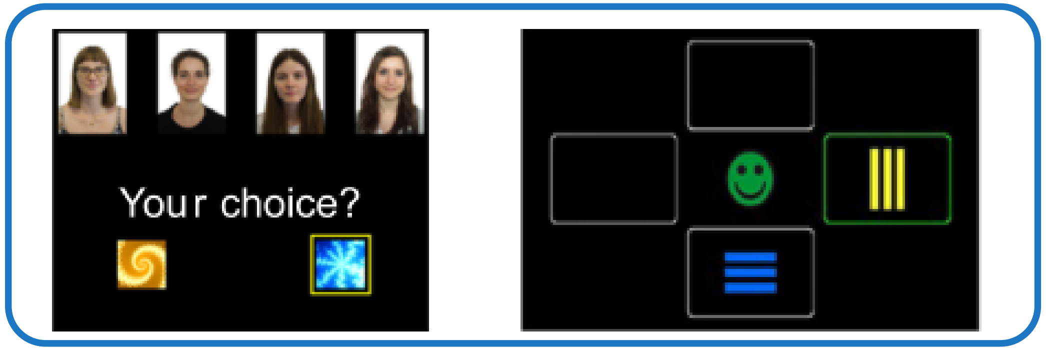
UNIVERSITY OF BIRMINGHAM

CHBH
CENTRE FOR HUMAN BRAIN HEALTH

My research:

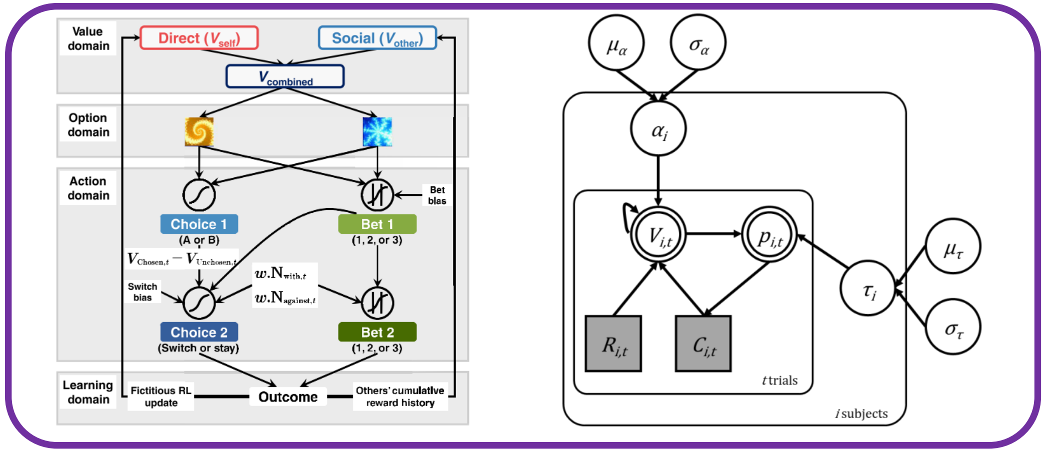
- I ask people to make decisions

Computation



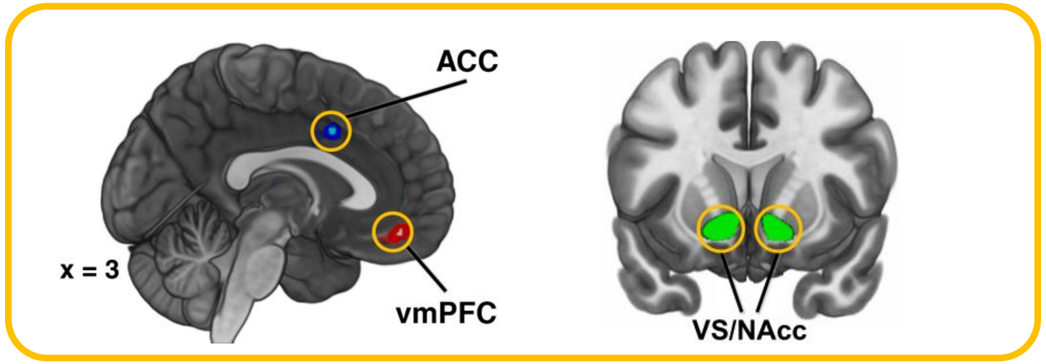
- I build computational models

Algorithm



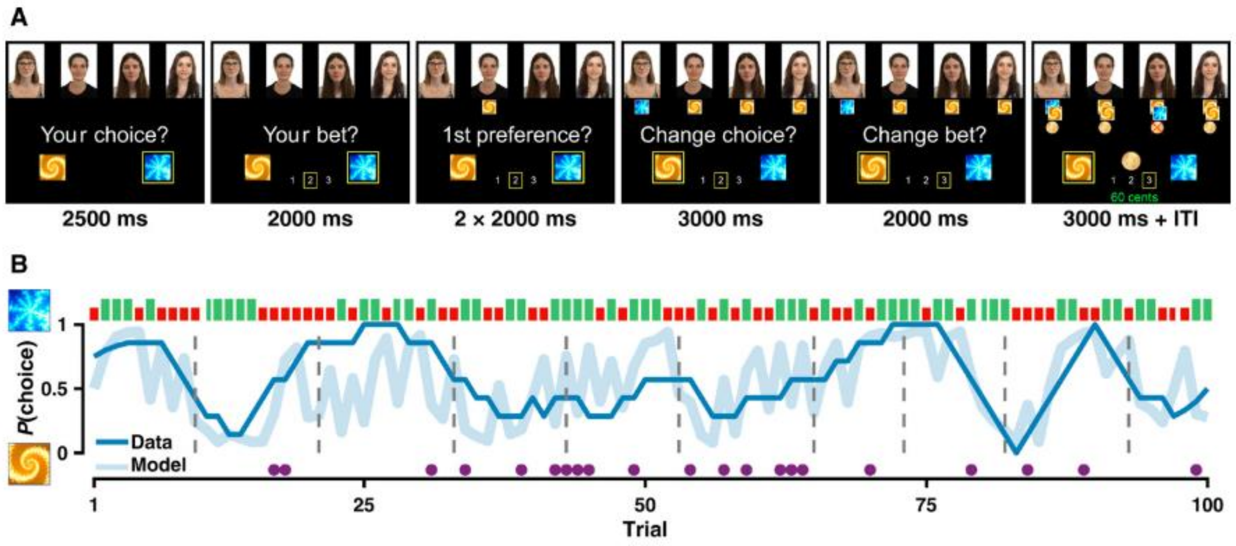
- I examine neural mechanisms

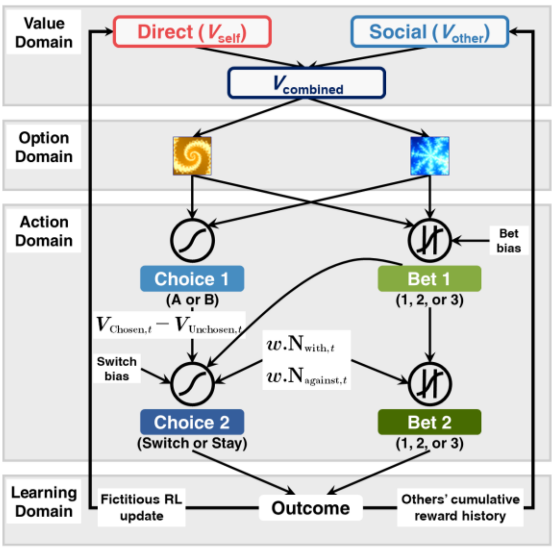
Implementation



About me

- My journey through computational modeling
  - Started with MLE (@fminsearch in Matlab)
  - Switched to Bayesian: first JAGS, then Stan
  - Why switching to Stan? Look at Fig. 2A and Table S3 in [Zhang & Gläscher (2020)]()





$$
\begin{aligned}
\mathbb{V}_{self,t} &= [V_{self,t}(\mathrm{A}), V_{self,t}(\mathrm{B})] \\
\mathbb{V}_{other,t} &= [V_{other,t}(\mathrm{A}), V_{other,t}(\mathrm{B})] \\
\mathbb{V}_{t} &= \beta_{vself}\,\mathbb{V}_{self,t} + \beta_{vother}\,\mathbb{V}_{other,t} \\
C1_t &\sim Categorical(Softmax(\mathbb{V}_t)) \\
U_{bet1,t} &= \beta_{bias_{b1}} + \beta_{vdiff_{b1}}(V_{chosen,C1,t} - V_{unchosen,C1,t}) \\
B1_t &\sim OrderedLogistic(U_{bet1,t} \mid \theta) \\
w.\mathrm{N}_{against,t} &= \frac{\sum_{s=1}^{K} w_{s,t}}{\sum_{s=1}^{4} w_{s,t}}, K = 0, 1, \ldots, 4 \\
w.\mathrm{N}_{with,t} &= \frac{\sum_{s=1}^{4-K} w_{s,t}}{\sum_{s=1}^{4} w_{s,t}} \\
V_t(\text{switch}) &= \beta_{bias_{c2}} + \beta_{vdiff_{c2}}(V_{chosen,C1,t} - V_{unchosen,C1,t}) + \beta_{against}\,w.\mathrm{N}_{against,t} \\
C2 &\sim Bernoulli(V_t(\text{switch})) \\
U_{bet2,t} &= \begin{cases} U_{bet1,t} + \beta_{with_{stay}}\,w.\mathrm{N}_{with,t} + \beta_{against_{stay}}\,w.\mathrm{N}_{against,t}, & \text{if } C1 = C2 \\ U_{bet1,t} + \beta_{with_{switch}}\,w.\mathrm{N}_{with,t} + \beta_{against_{switch}}\,w.\mathrm{N}_{against,t}, & \text{if } C1 \neq C2 \end{cases} \\
B2_t &\sim OrderedLogistic(U_{bet2,t} \mid \theta) \\
\Phi(x) &= \frac{1}{1+e^{-x}} \\
\delta_{self,chosen,C2,t} &= R_{self,t} - V_{self,chosen,C2,t} \\
\delta_{self,unchosen,C2,t} &= -R_{self,t} - V_{self,unchosen,C2,t} \\
V_{self,chosen,C2,t+1} &= V_{self,chosen,C2,t} + \alpha\delta_{self,chosen,C2,t} \\
V_{self,unchosen,C2,t+1} &= V_{self,unchosen,C2,t} + \alpha\delta_{self,unchosen,C2,t}
\end{aligned}
$$

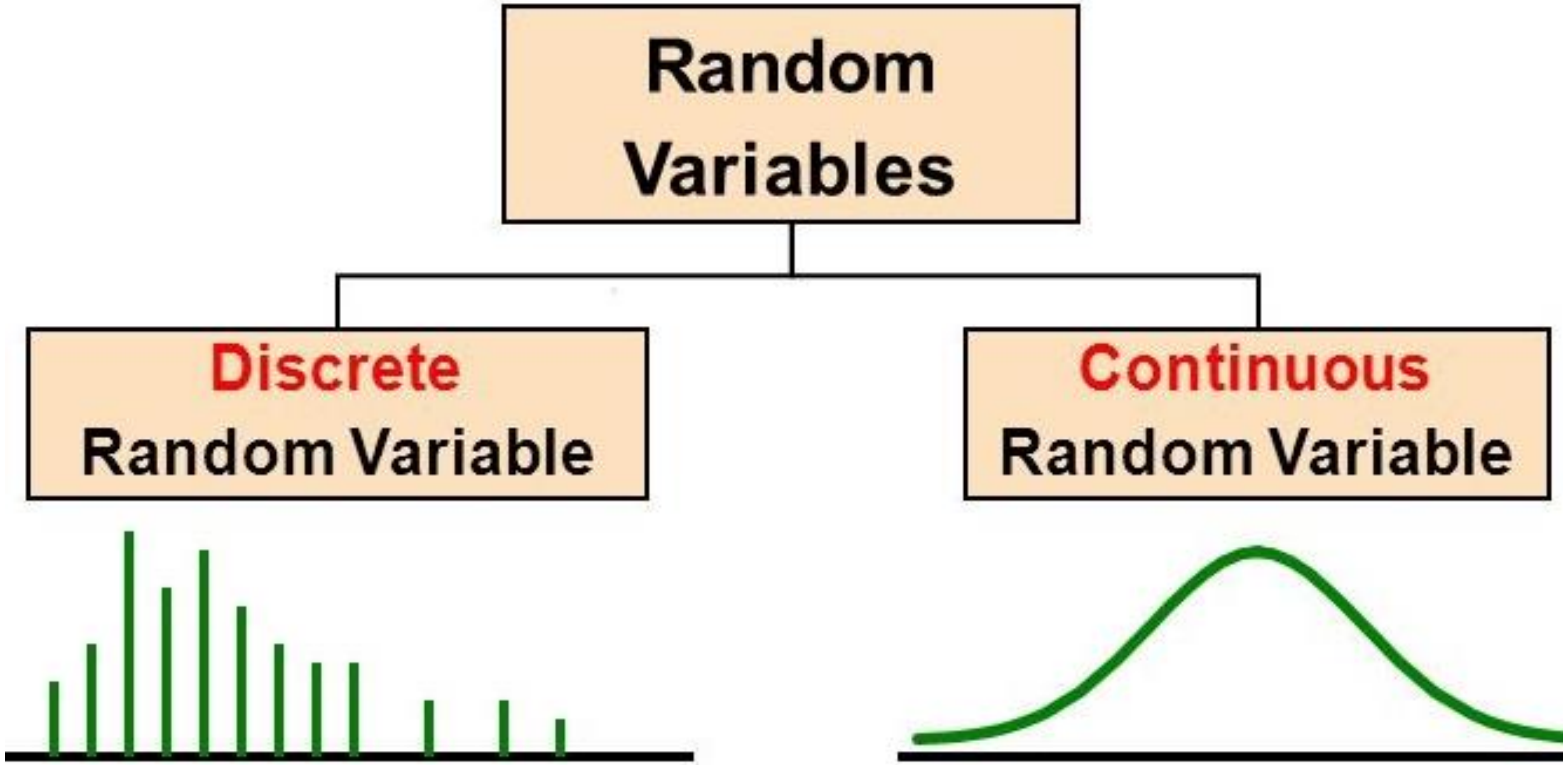
Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the simple Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

Probability Functions

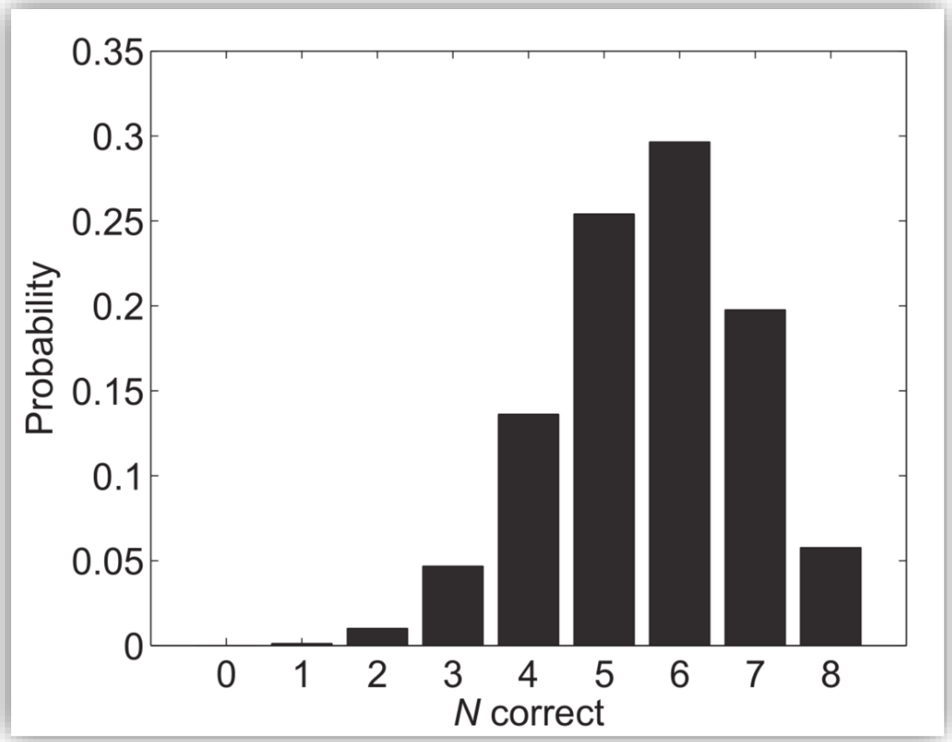



μ ψ
MATHPSYCH /
ICCM / EMPG
2023
18 - 21 July

Probability Functions

discrete variable – we talk about mass

- Run a test of 8 questions, and record each student's correct responses
- Count and plot the # of correct answers
- Then divide those counts by the total number of students
- The lengths of these bars sum up to 1



Lewandowsky & Farrell (2011)

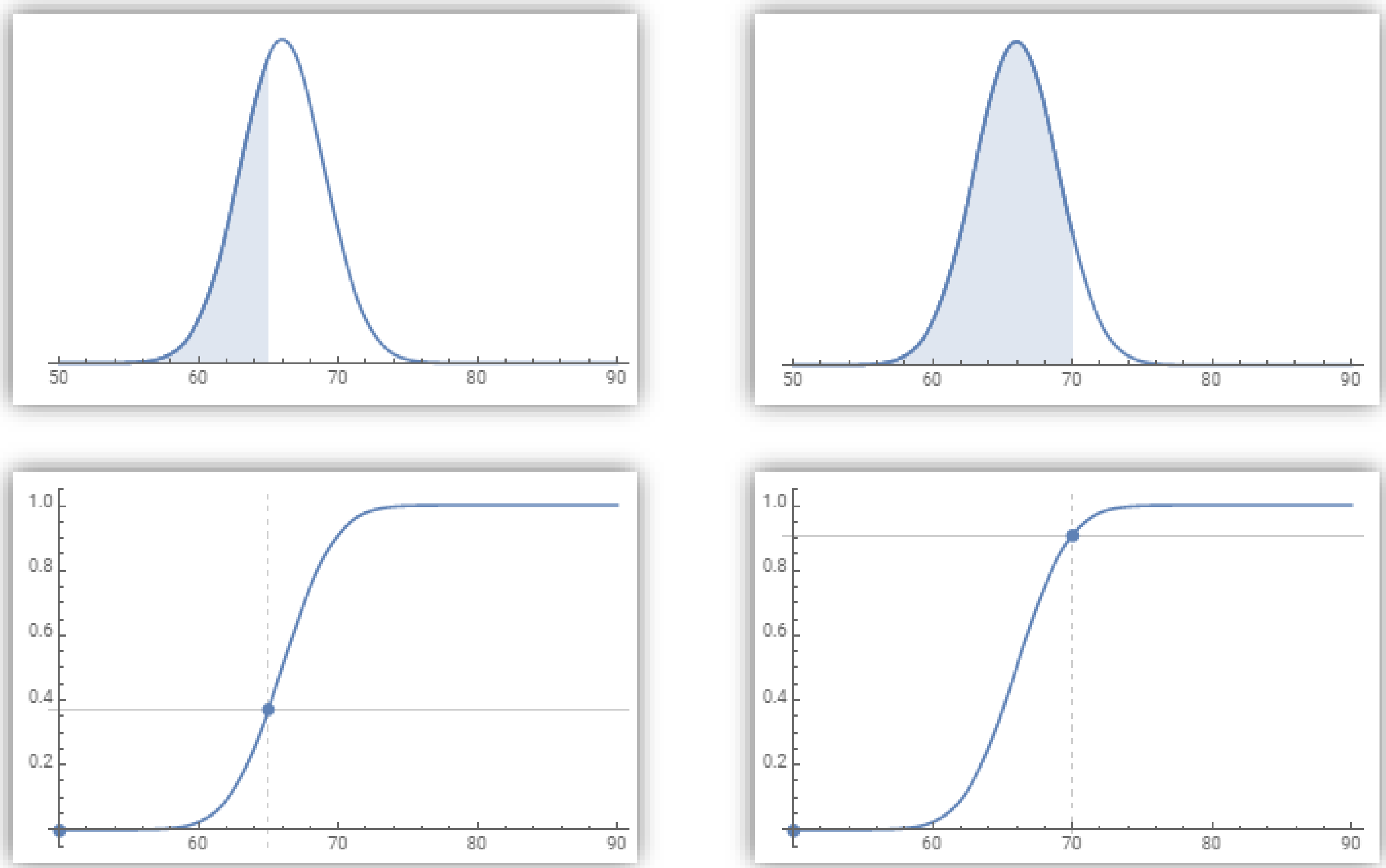
Probability Functions

continuous variable – we talk about density

probability density function
(PDF)

The entire area under the curve
of PDF is 1.

cumulative distribution function
(CDF)



https://demonstrations.wolfram.com/ConnectingTheCDFAndThePDF/

Joint, Marginal and Conditional Probability

Joint Probability

$p(A, B) = p(B, A)$

- e.g., $p(\text{rain, cold})$: $p(\text{rain})$ AND $p(\text{cold})$

Marginal Probability

$p(A)$ – 'p of A irrespective of B'

- e.g., $p(\text{rain})$: $p(\text{rain, cold}) + p(\text{rain, not cold})$

Conditional Probability

$p(A|B)$ – 'p of A given B' – event B is fixed, not uncertainty

$p(A,B) = p(A|B)p(B)$

- e.g., $p(\text{rain, cold}) = p(\text{rain}|\text{cold})p(\text{cold})$

Example: discrete

snow

X

		1	0
cold Y	1	0.5	0.1
	0	0.1	0.3

Joint probability :

$$P(X=0,Y=1)=$$

$$\sum_{x,y} P(X=x,Y=y)=1$$

Marginal probability :

$$P(Y=1)=$$

$$P(X=0)=$$

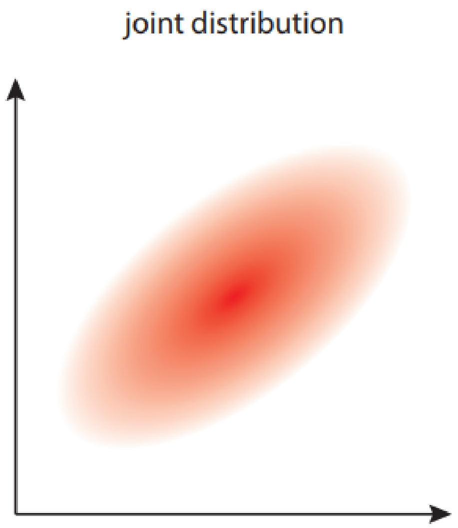
$$P(X=x)=\sum_{y} P(X=x,Y=y)$$

Conditional probability :

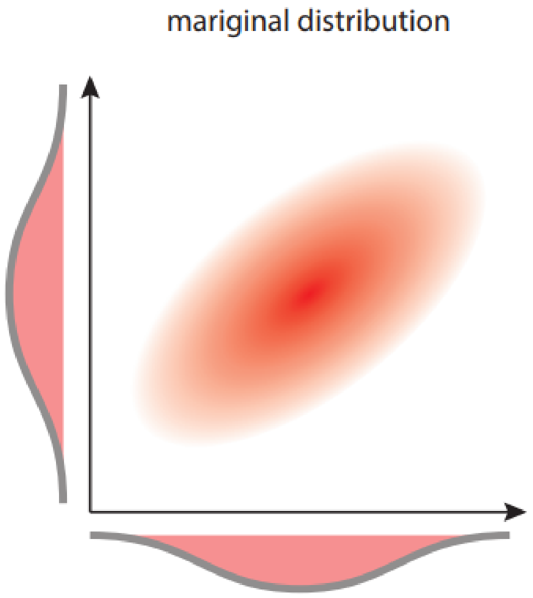
$$P(X=1|Y=1)=$$

$$P(X=x|Y=y)=\frac{P(X=x,Y=y)}{P(Y=y)}$$
$$=\frac{P(X=x,Y=y)}{\sum_x P(X=x,Y=y)}$$

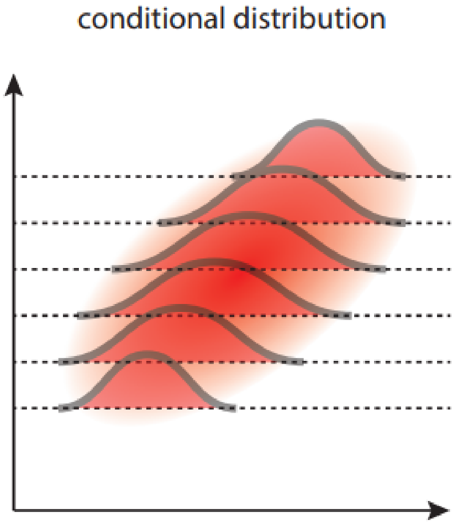
Example: continuous



The "co-distribution" of x and y.



The density of x- (or y-) values, without knowing the other's value.



The probability distribution of x, given that we know the value of y.

Credit: [Eric Ma]

Bayes' theorem

$$p(A,B) = p(B,A)$$

$$p(A,B) = p(A|B)p(B)$$

$$p(B,A) = p(B|A)p(A)$$

$$p(A|B)p(B) = p(B|A)p(A)$$

$$\boxed{p(A \mid B) = \frac{p(B \mid A)\, p(A)}{p(B)}}$$

Linking Data and Parameter

θ $\quad\quad\quad$ D

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

Linking Data and Parameter

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

Linking Data and Parameter

Likelihood
How plausible is the data given our parameter is true?

Prior
How plausible is our parameter before observing the data?

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

Posterior
How plausible is our parameter given the observed data?

Evidence
How plausible is the data under all possible parameters?

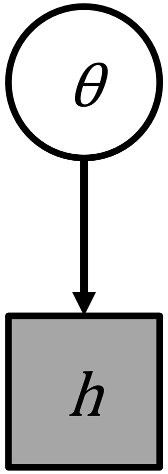
Bernoulli Model

- You are interested in if a coin is biased.
- You will flip the coin.
- You will record whether it comes up a head (h) or a tail (t).
- You might observe 15 heads out of 20 flips.
- What is your degree of belief about the biased parameter ϑ?

Kruschke (2015)

Bernoulli Model

$$p(h \mid \theta) = \theta^{h}(1-\theta)^{1-h}$$



$$\theta \sim \text{Uniform}(0, 1)$$

$$h \sim \text{Bernoulli}\ (\theta)$$

	continuous	discrete
unobserved	θ	δ
observed	y	N

Steps of (Bayesian) Modeling?

A data story

Think about how the data might arise.

It can be *descriptive* or even *causal*.

Update

Educate your model by feeding it with data.

Bayesian Update:

- update the prior, in light of data, to produce posterior
- the updated posterior then becomes the prior of next update

Evaluate

Compare model with reality.

Revise your model.

McElreath (2016)

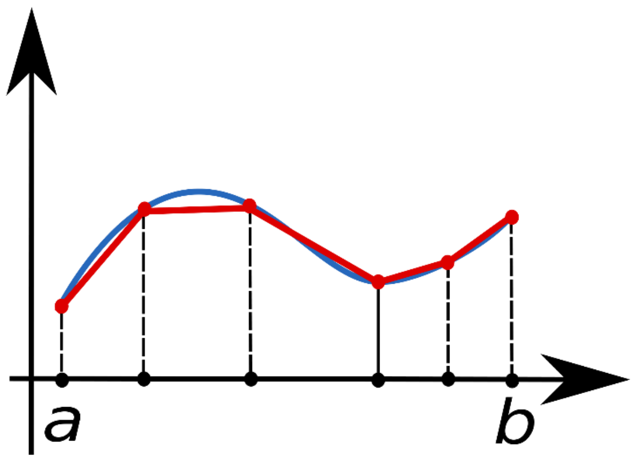
Solve it by Grid Approximation

discrete parameters

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\boxed{\sum_{\theta^*} p(D \mid \theta^*)\, p(\theta^*)}}$$

continuous parameters

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\boxed{\int p(D \mid \theta^*)\, p(\theta^*)\, d\theta^*}}$$



What if I have multiple parameters?

grid approximation for
2 parameters?
5 parameters?
10 parameters?

$$p(\theta \mid D) = \frac{p(D \mid \theta)\, p(\theta)}{\boxed{\int p(D \mid \theta^*)\, p(\theta^*)\, d\theta^*}}$$

$$p(data) = \int_{\text{All}\theta_1} \int_{\text{All}\theta_2} p(data, \theta_1, \theta_2)\,\mathrm{d}\theta_1 \mathrm{d}\theta_2$$

$$p(data) = \int_{\mu_1} \int_{\sigma_1} \cdots \int_{\mu_{100}} \int_{\sigma_{100}} \underbrace{p(data \mid \mu_1, \sigma_1, \ldots, \mu_{100}, \sigma_{100})}_{\text{likelihood}} \times \underbrace{p(\mu_1, \sigma_1, \ldots, \mu_{100}, \sigma_{100})}_{\text{prior}}$$
$$\mathrm{d}\mu_1 \mathrm{d}\sigma_1 \ldots \mathrm{d}\mu_{100} \mathrm{d}\sigma_{100},$$

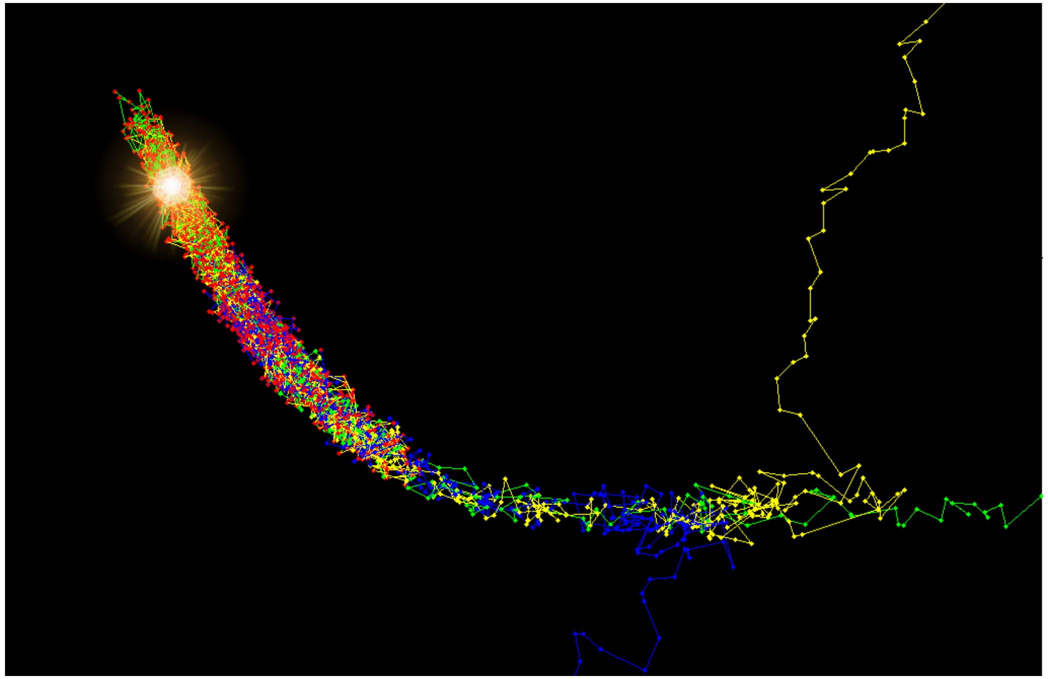
- Analytical solutions (often does not exist)
- Grid approximation (takes too long)

$$\boxed{p(\theta \mid D) \propto p(D \mid \theta)\, p(\theta)}$$

- solution: Markov Chain Monte Carlo

MCMC Sampling Algorithms

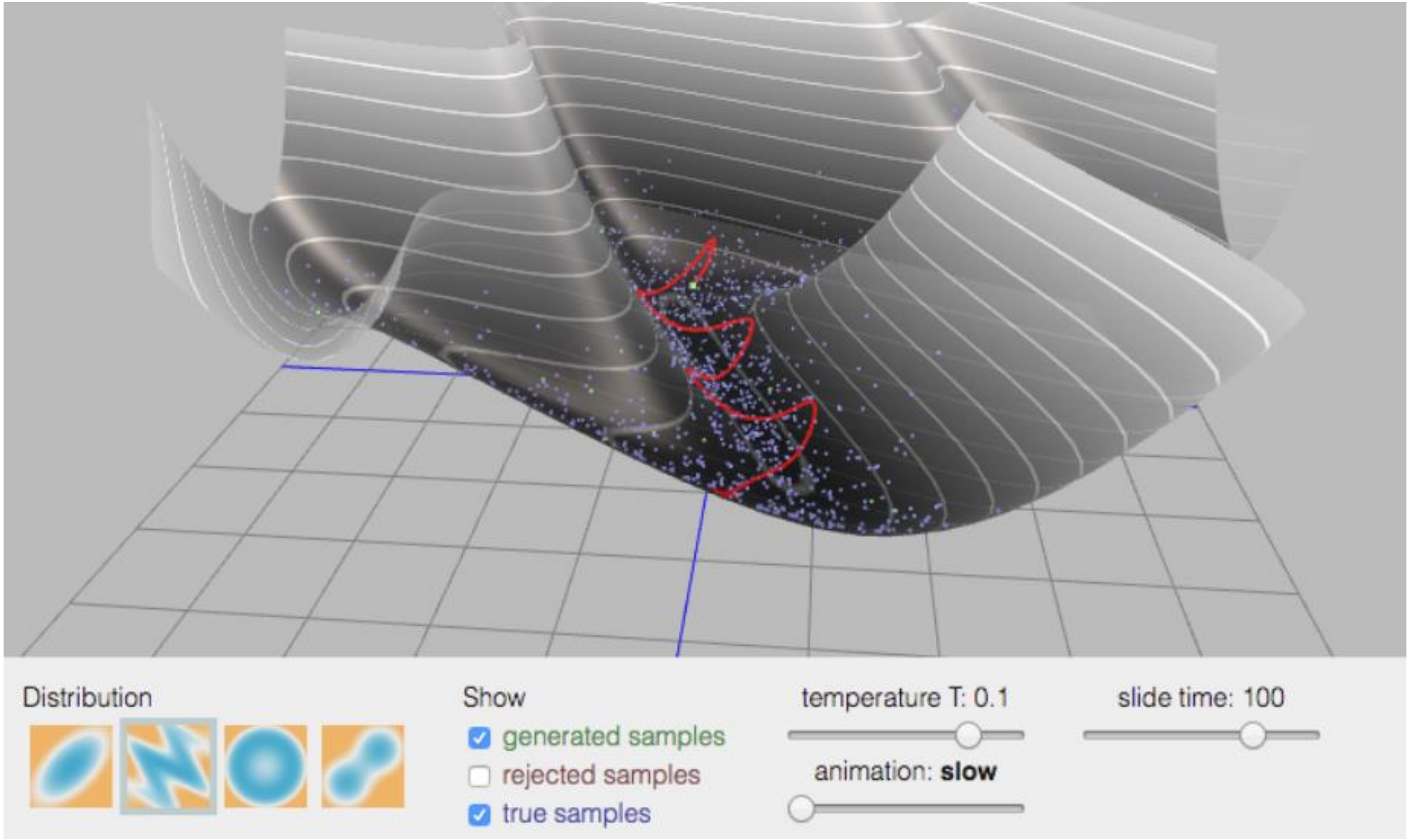
- Rejection sampling
- Importance sampling
- Metropolis algorithm
- Gibbs sampling → JAGS
- HMC sampling*



Stan!

*: also PyMC3 PyMC3

build some intuition



https://arogozhnikov.github.io/2016/12/19/markov_chain_monte_carlo.html

Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the simple Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

Is Stan popular?

Stan: A Probabilistic Programming Language

Carpenter, Bob; Gelman, Andrew; Hoffman, Matthew D.; Lee, Daniel; Goodrich, Ben; Betancourt, Michael; Brubaker, Marcus A.; Guo, Jiqiang; Li, Peter; Riddell, Allen

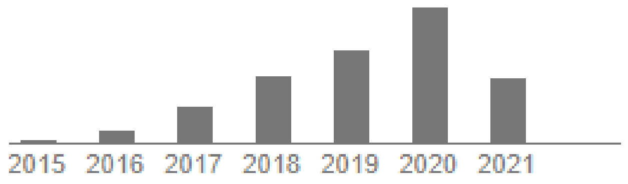
Grantee Submission, Journal of Statistical Software v76 n1 p1-32 Jan 2017

Stan is a probabilistic programming language for specifying statistical models. A Stan program imperatively defines a log probability function over parameters conditioned on specified data and constants. As of version 2.14.0, Stan provides full Bayesian inference for continuous-variable models through Markov chain Monte Carlo methods such as the No-U-Turn sampler, an adaptive form of Hamiltonian Monte Carlo sampling. Penalized maximum likelihood estimates are calculated using optimization methods such as the limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm. Stan is also a platform for computing log densities and their gradients and Hessians, which can be used in alternative algorithms such as variational Bayes, expectation propagation, and marginal inference using approximate integration. To this end, Stan is set up so that the densities, gradients, and Hessians, along with intermediate quantities of the algorithm such as acceptance probabilities, are easily accessible. Stan can be called from the command line using the "cmdstan" package, through R using the "rstan" package, and through Python using the "pystan" package. All three interfaces support sampling and optimization-based inference with diagnostics and posterior analysis. "rstan" and "pystan" also provide access to log probabilities, gradients, Hessians, parameter transforms, and specialized plotting.

Descriptors: Programming Languages, Probability, Bayesian Statistics, Monte Carlo Methods, Statistical Inference, Maximum Likelihood Statistics, Computation, Statistical Distributions, Computer Software

2021 Summer

Cited by 3989



[PDF] **JAGS**: A program for analysis of Bayesian graphical models using Gibbs sampling

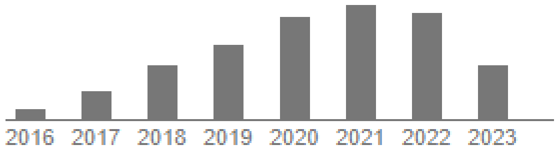
M Plummer - Proceedings of the 3rd international workshop on …, 2003 - ci.tuwien.ac.at

JAGS is a program for Bayesian Graphical modelling which aims for compatibility with Classic BUGS. The program could eventually be developed as an R package. This article explains the motivations for this program, briefly describes the architecture and then …

☆ 99 Cited by 4543 Related articles All 8 versions »

2023 Summer

Cited by 6594



[PDF] **JAGS**: A program for analysis of Bayesian graphical models using Gibbs sampling

M Plummer - Proceedings of the 3rd international workshop on …, 2003 - r-project.org

… **JAGS** is a program for Bayesian Graphical modelling which aims for compatibility … **JAGS** and so avoid having to write a new program for each application. A second motivation for **JAGS** …

☆ Save 99 Cite Cited by 6277 Related articles All 7 versions »

b~r+(m|s)

brms: An R package for Bayesian multilevel **models** using Stan

PC Bürkner - Journal of statistical software, 2017 - jstatsoft.org

… The **brms** package does not fit **models** itself but uses Stan on the back-end. Accordingly, all samplers implemented in Stan can be used to fit **brms models**. Currently, these are the static …

☆ Save 99 Cite Cited by 5534 Related articles All 29 versions »

Who are using Stan?

retrieved in 05/2021, [Web of Science]

Why Use Stan?

vs. BUGS / JAGS

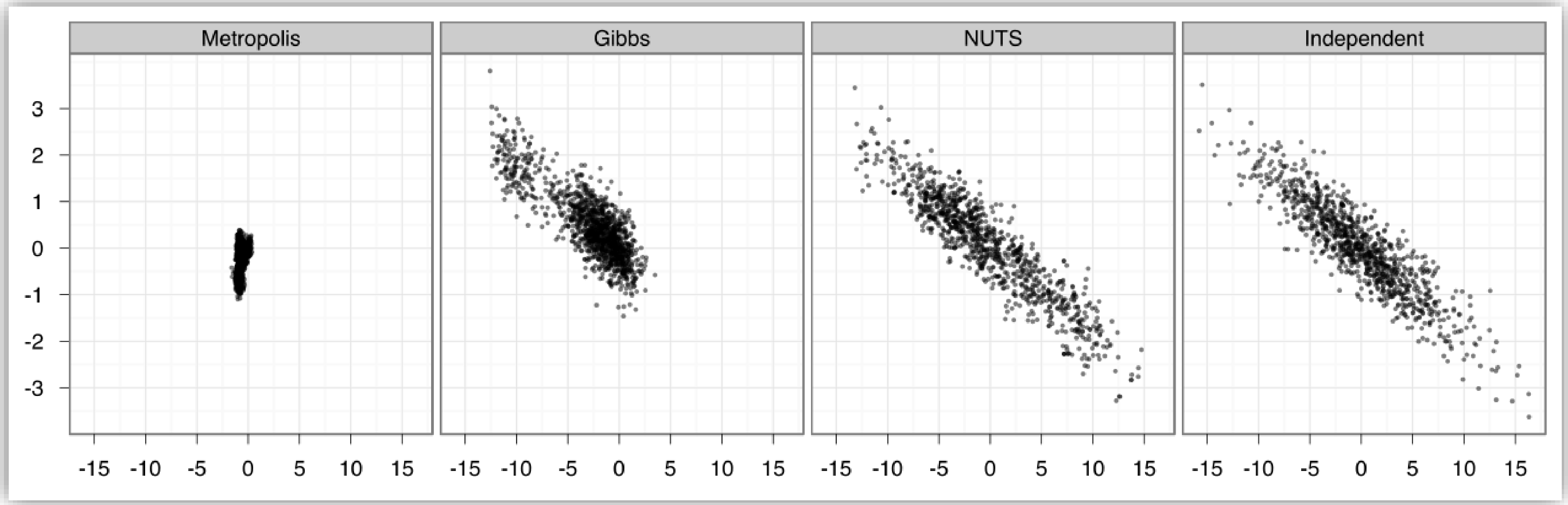
- Less spatial correlation → effective samples
- Time to converge and per effective sample size:

  1 - ∞ times faster
- Memory usage: 1–10%
- Language features
  - variable overwrite: `a = 4, then a = 5`
  - formal control flow (same as R)
  - full support of vectorizing & matrix calculation

http://mc-stan.org/
https://discourse.mc-stan.org/t/selling-stan/3693/6

HMC vs. Gibbs and Metropolis

Hamilton MC (HMC) implements No-U-Turn Sampler (NUTS)



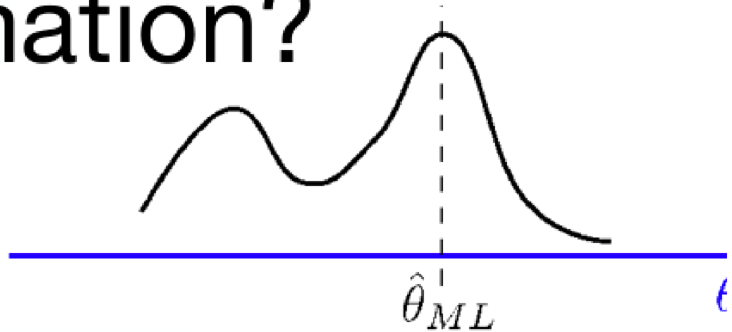
- Two dimensions of highly correlated 250-dim normal
- 1,000,000 draws from Metropolis and Gibbs (thin to 1000)
- 1,000 draws from NUTS; 1000 independent draws

Stan and Other Platforms

Data

Samples

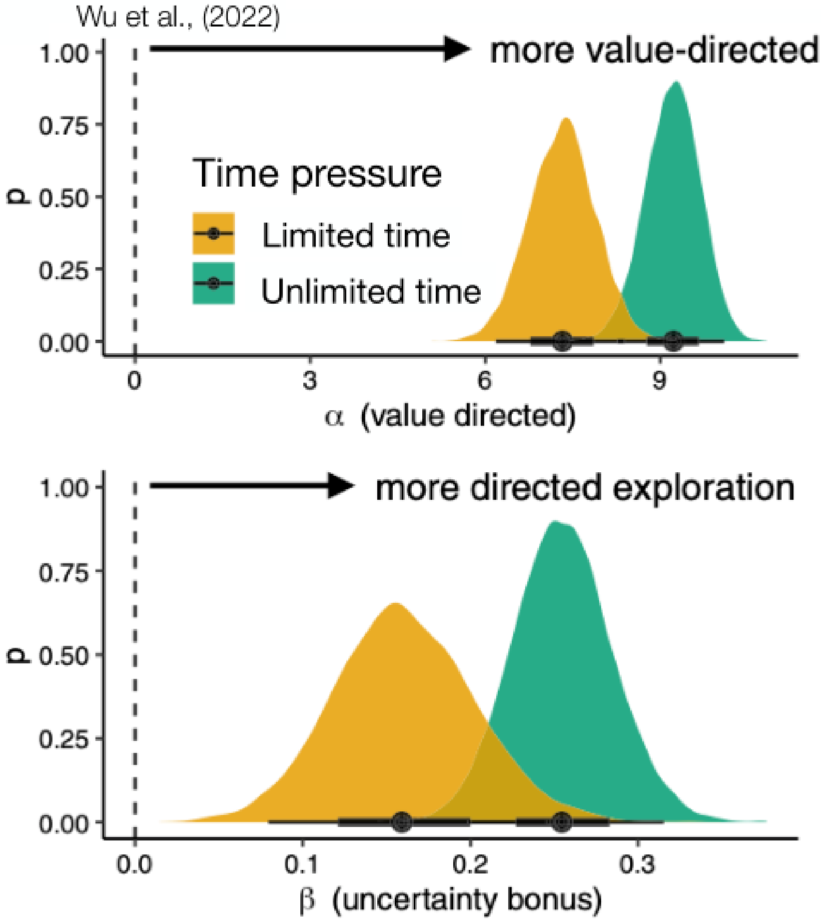
Why Bayesian model estimation?



1. Not just a point estimate, but an entire **probability distribution over parameters**

2. Rather than only assuming participants are independent samples, we can model **hierarchical relationships**

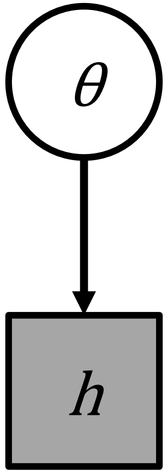
3. Naturally avoid overfitting through **Bayesian Occam's Razor**, since we evaluate the model across the entire range of parameters



But, any cons?

- HMC provides huge improvements in computational efficiency, but mathematical foundations are more difficult to follow, at least sometimes.
- Stan cannot sample from the posterior distribution of discrete parameters (e.g., [1, 2]).
  - → with additional effort, it can be achieved through marginalisation, see the Stan User Manual.
- There are a few unwritten tips and tricks.
  - Practise makes perfect and follow discussions on the forum.
- Block-based language → might seem rigid in the first place.

A quick look at Stan



$$\theta \sim \text{Uniform}(0, 1)$$

$$h \sim \text{Bernoulli}\ (\theta)$$

```
data {
  int<lower=0> N;
  int<lower=0,upper=1> flip[N];
}

parameters {
  real<lower=0,upper=1> theta;
}

model {
  flip ~ bernoulli(theta);
}
```

Stan Language

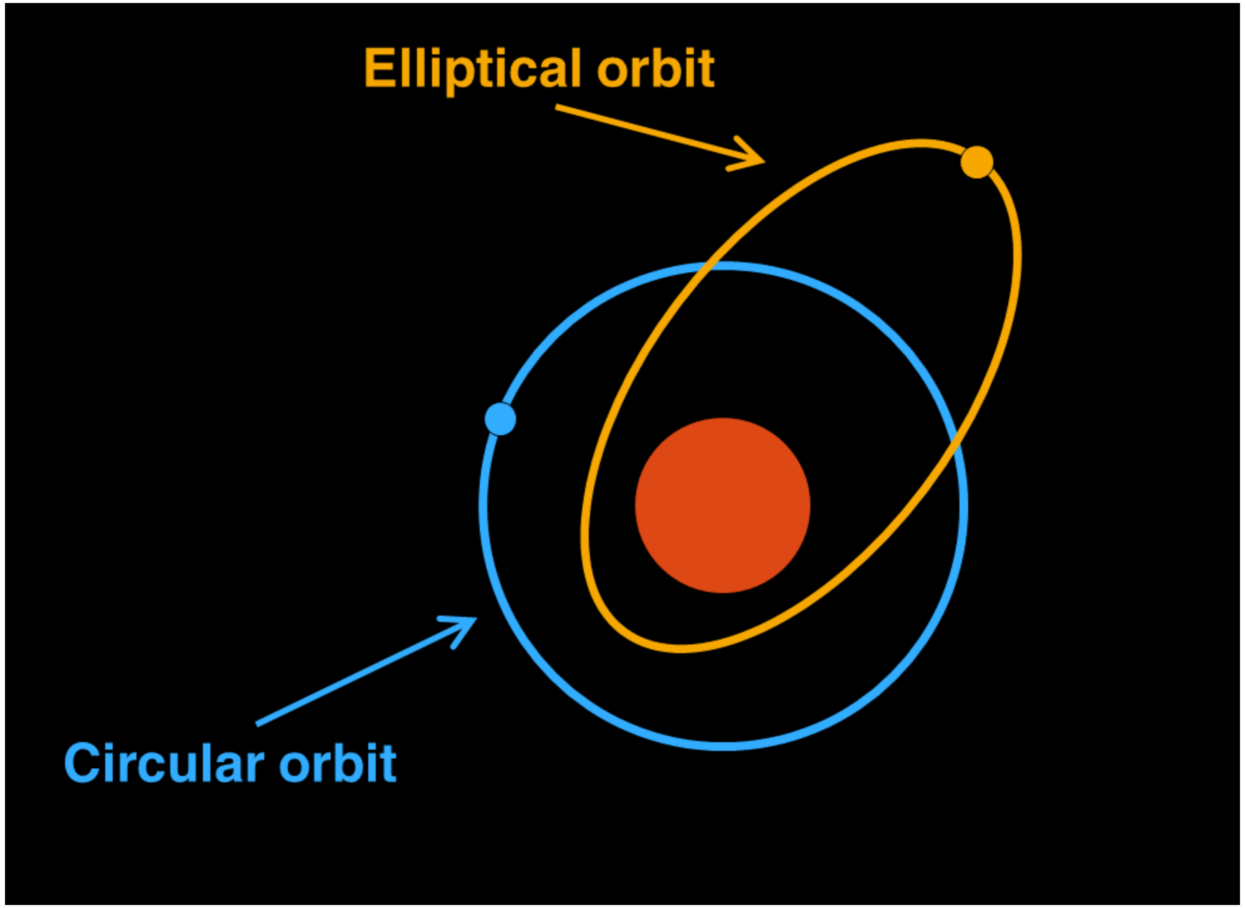
```
data {
 //... read in external data...
}

transformed data {
 //... pre-processing of data ...
}

parameters {
 //... parameters to be sampled by HMC ...
}

transformed parameters {
 //... pre-processing of parameters ...
}

model {
 //... statistical/cognitive model ...
}

generated quantities {
 //... post-processing of the model ...
}
```

Outline

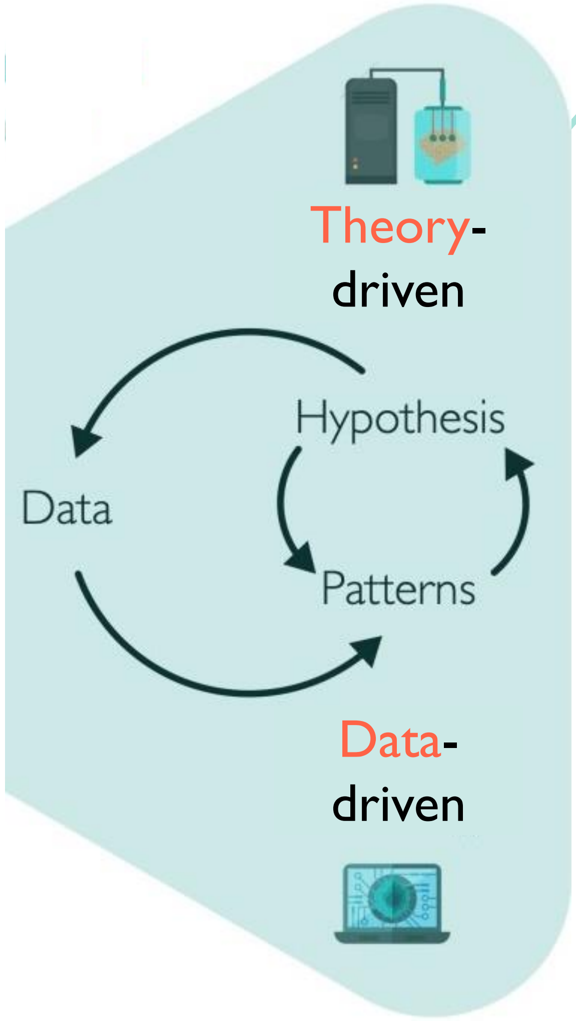
- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the simple Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

The idea of computational modeling is never new

Scientists use mathematical models to approximate certain processes (physical or mental), in order to explain and to predict.



Two flavours of computational modeling



Provide mechanistic explanation

Generate prediction/ classification

Construct

RDoC
Research Domain Criteria Initiative

- Reward,
- Learning,
- Self v. Other,
- etc.

Computational modeling

On the Role of Theory and Modeling in Neuroscience

Daniel Levenstein,[1] Veronica A. Alvarez,[2] Asohan Amarasingham,[3] Habiba Azab,[4] Zhe S. Chen,[5] Richard C. Gerkin,[6] Andrea Hasenstaub,[7] Ramakrishnan Iyer,[8] Renaud B. Jolivet,[9] Sarah Marzen,[10] Joseph D. Monaco,[11] Astrid A. Prinz,[12] Salma Quraishi,[13] Fidel Santamaria,[13] Sabyasachi Shivkumar,[14] Matthew F. Singh,[15] Roger Traub,[16] Farzan Nadim,[1,7*] Horacio G. Rotstein,[1,7*] and A. David Redish[18*]

How Computational Modeling Can Force Theory Building in Psychological Science

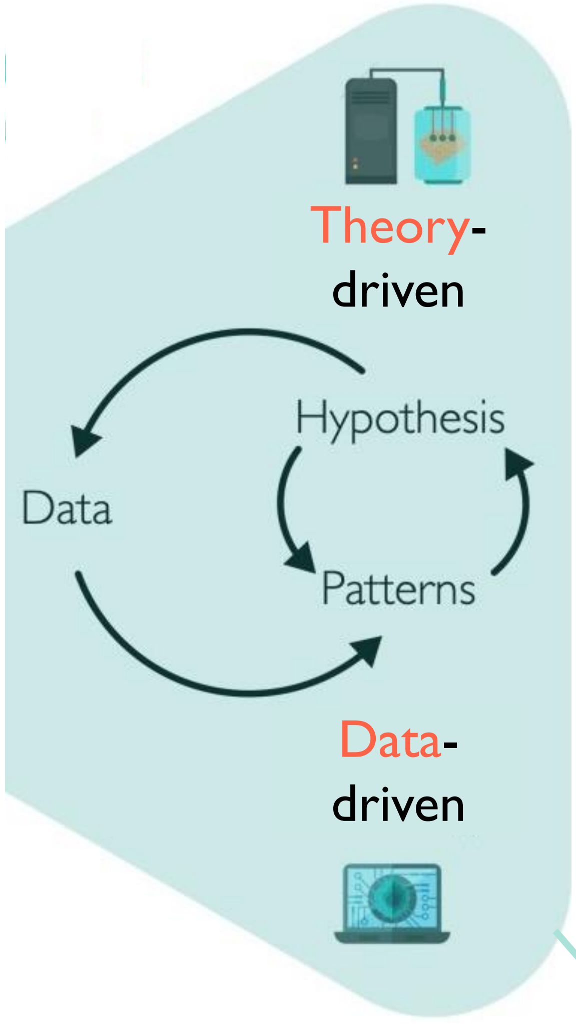
Olivia Guest, Andrea E. Martin

Building better theories

Clare Press [1, 2], Daniel Yon [1], Cecilia Heyes [3, 4]

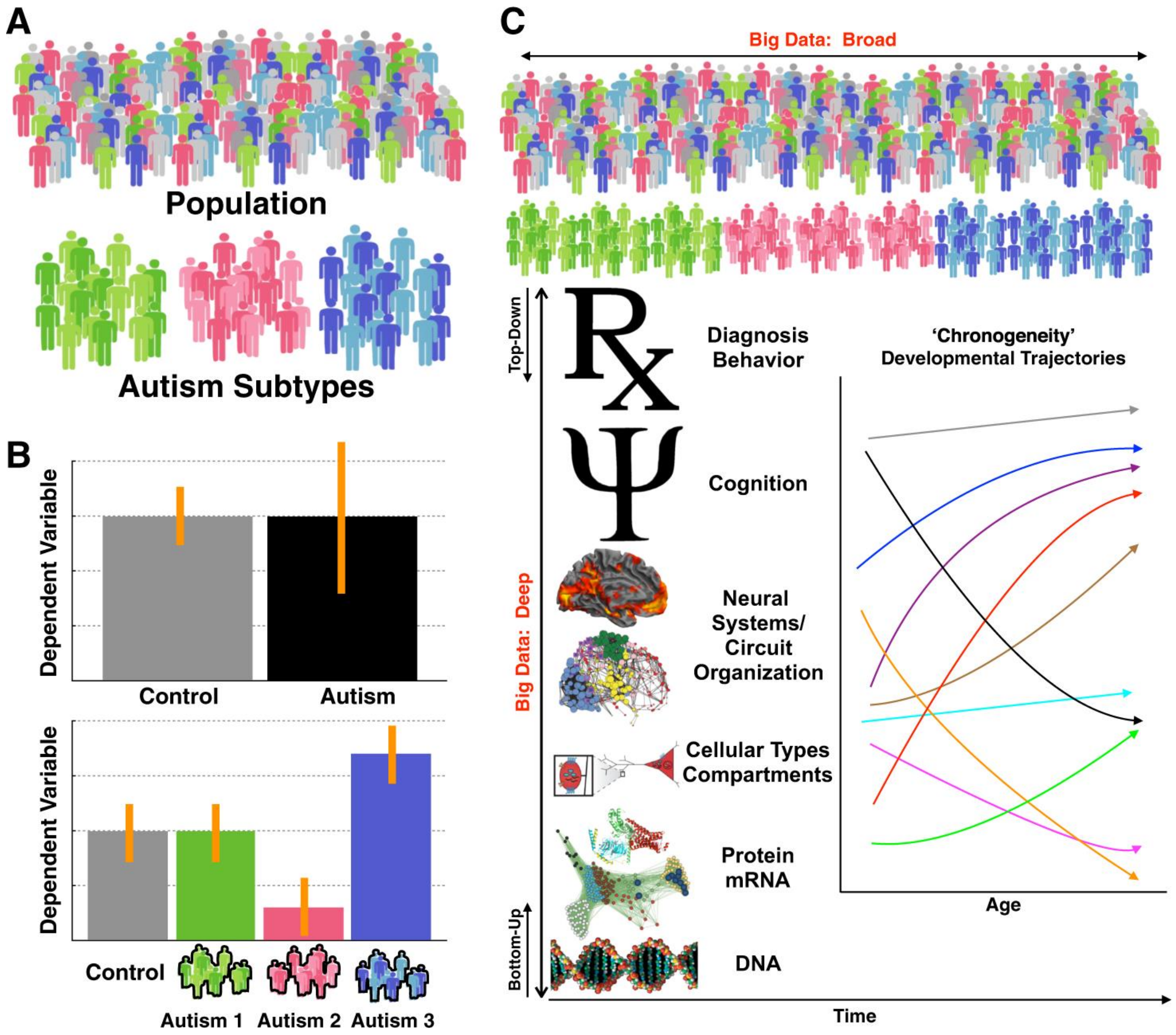
Huys et al., (2016); Paulus, (2017); Bennett et al., (2019); Gauld et al., (2021)

Two flavours of computational modeling



Provide **mechanistic explanation**

Generate **prediction/ classification**



Loth (2016); Lombardo et al. (2019)

Computational modeling

Cognition as information processing

Behavior

Output

Input

$$B = f(P, E)$$

$$y = f(x)$$

Person

Environment

'Some math happens'

Kurt Lewin, (1936)

Influential perspective: Marr's 3 levels of analysis

	LEVELS	
Computation	1	why (problem)
Algorithm	2	what (rules)
Implementation	3	how (physical)

Rolling resistance

Wheels

Climb

Behavior

Cognition

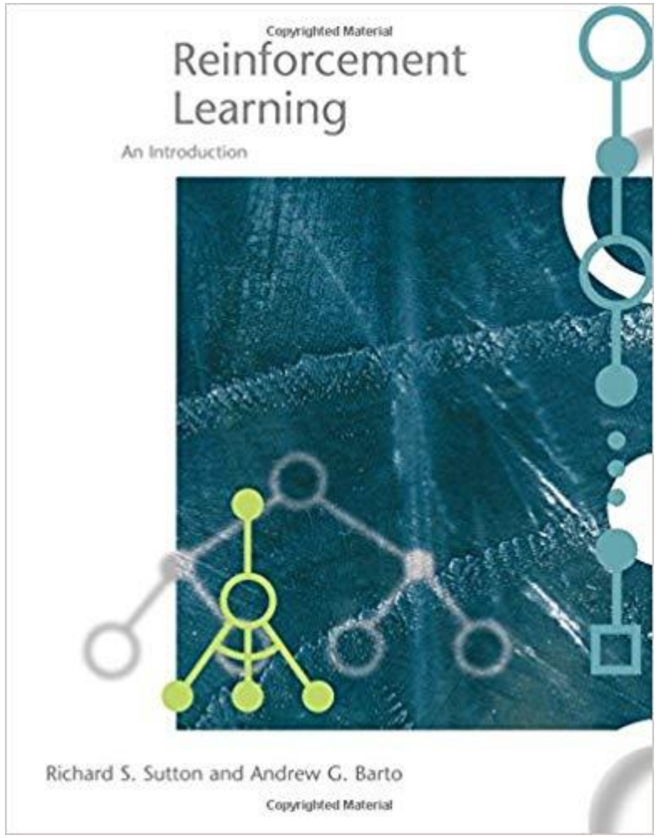
Brain

David Marr, (1982)

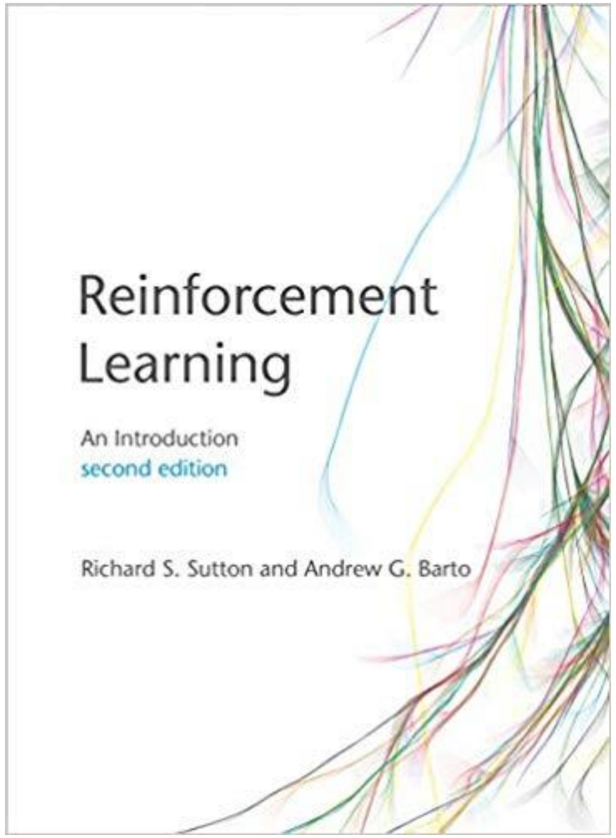
Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the simple Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

The very short history

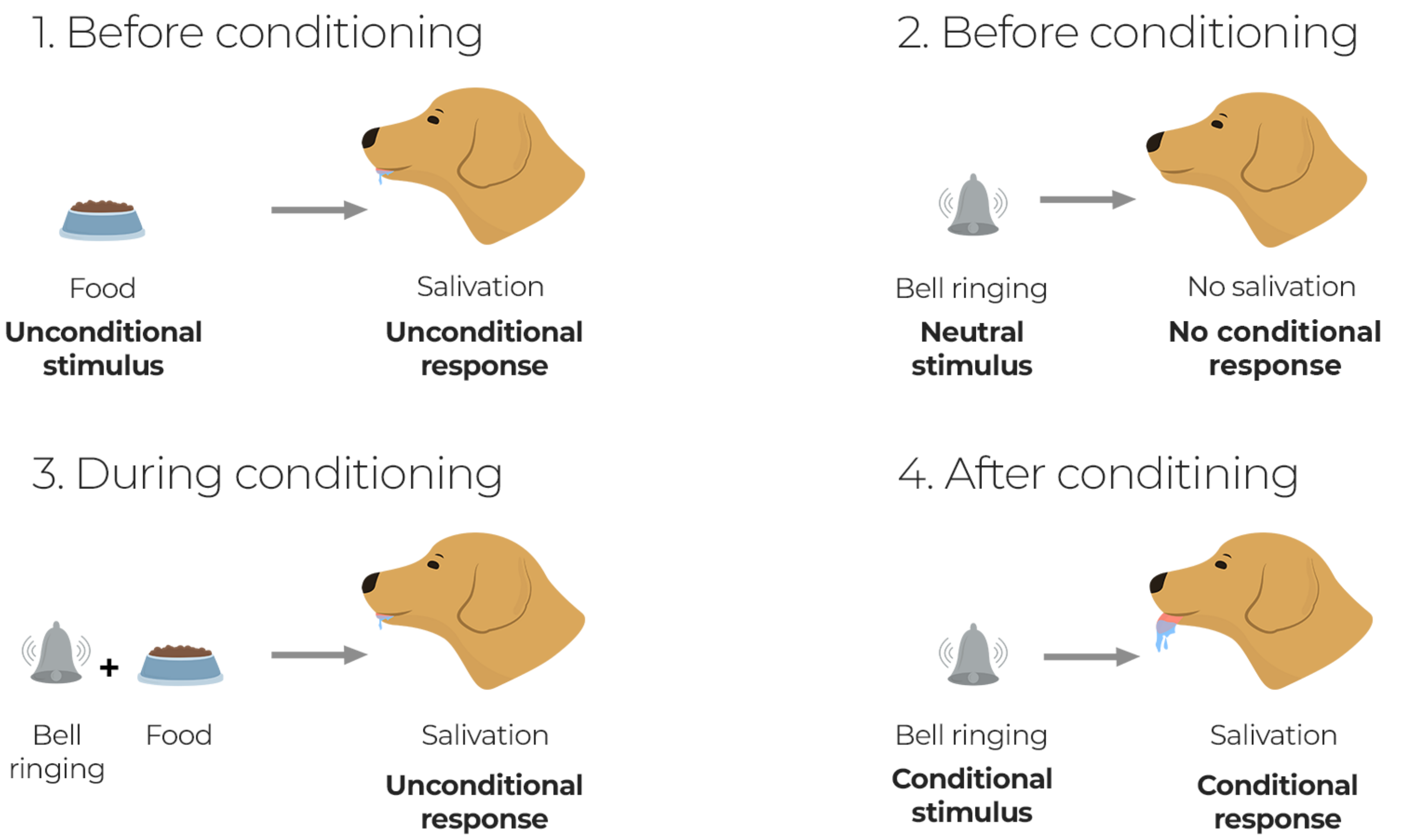


1998



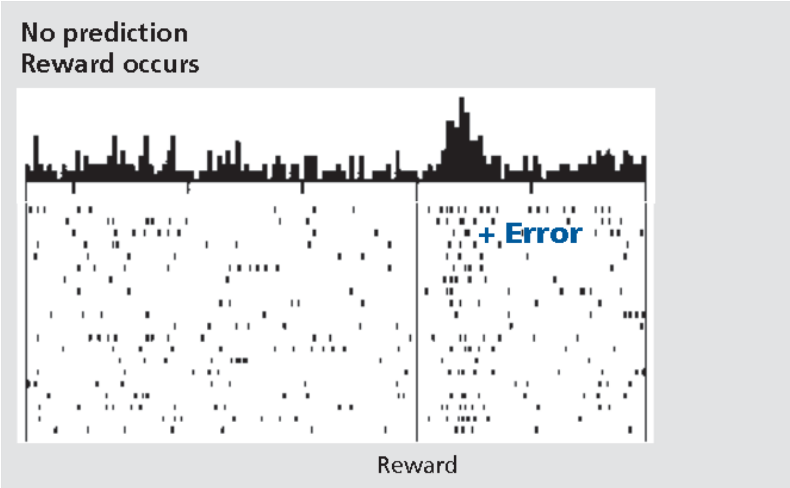
2018

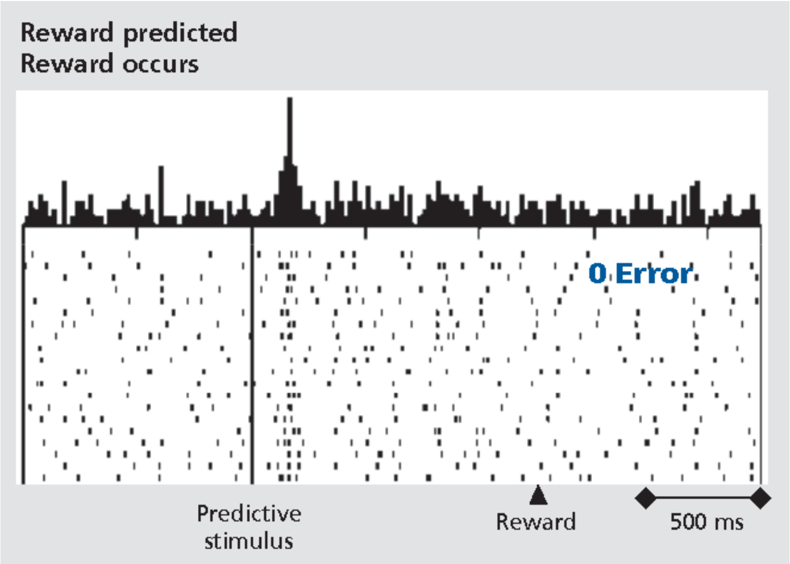
why is it relevant?

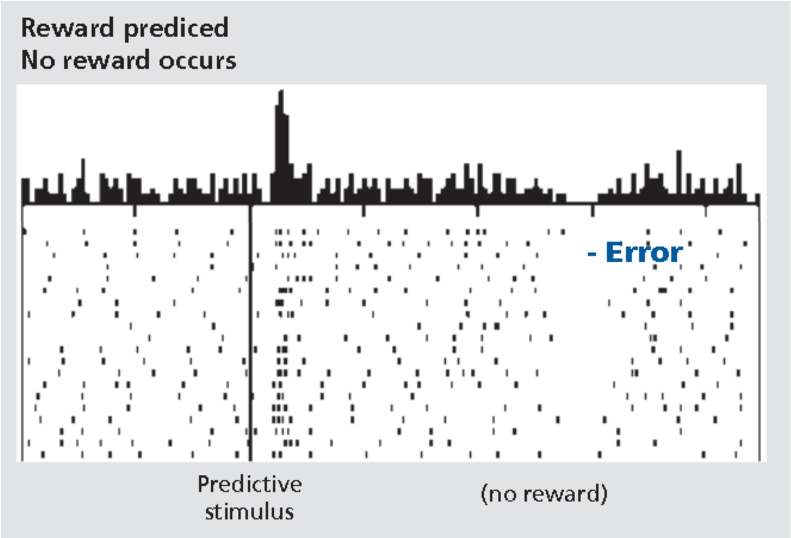


resource

Reward Prediction error







Predicted reward trials

A neural substrate of prediction and reward
W Schultz, P Dayan, PR Montague - Science, 1997 - science.org
The capacity to predict future events permits a creature to detect, model, and manipulate the causal structure of its interactions with its environment. Behavioral experiments suggest that …

Schultz, Dayan, & Montague (1997)

Simple reinforcement learning: 2-armed bandit task

a simple task often used in the laboratory:

- repeated choice between N options (N-armed bandit)
- …whose properties (reward amounts, probabilities) are learned through trial-and-error
- …with a goal in mind: maximize the overall reward

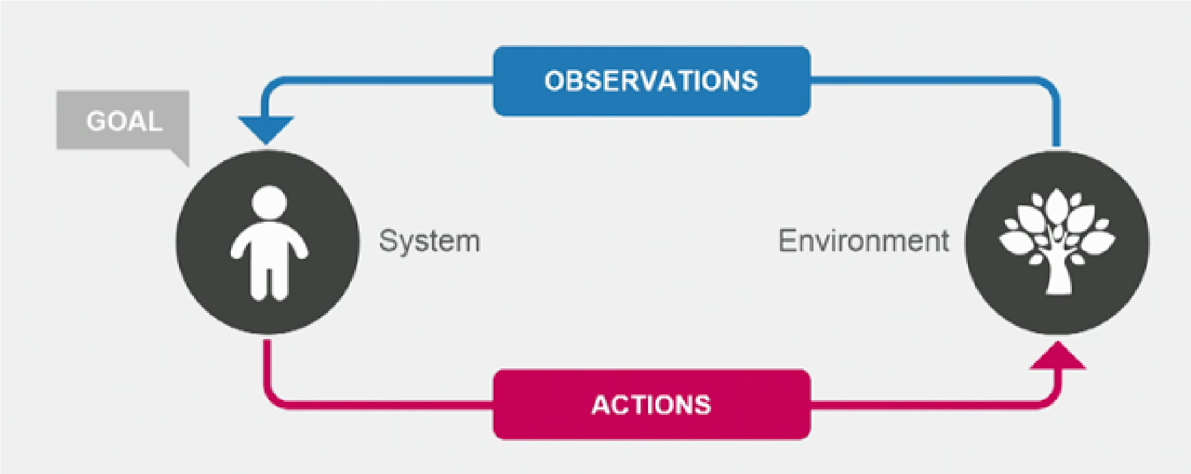
2-armed bandit task

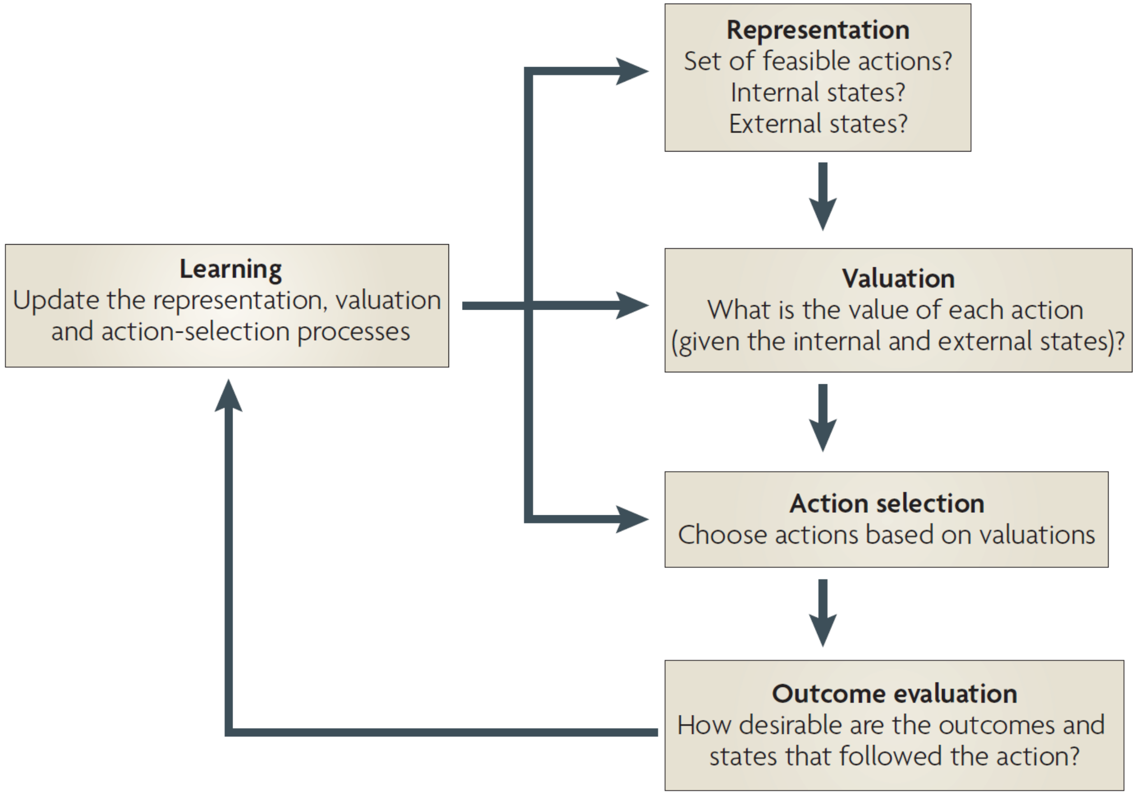
What can be your strategies:

1. predict the value of each deck
2. choose the best
3. learn from outcome to update predictions

(repeat)

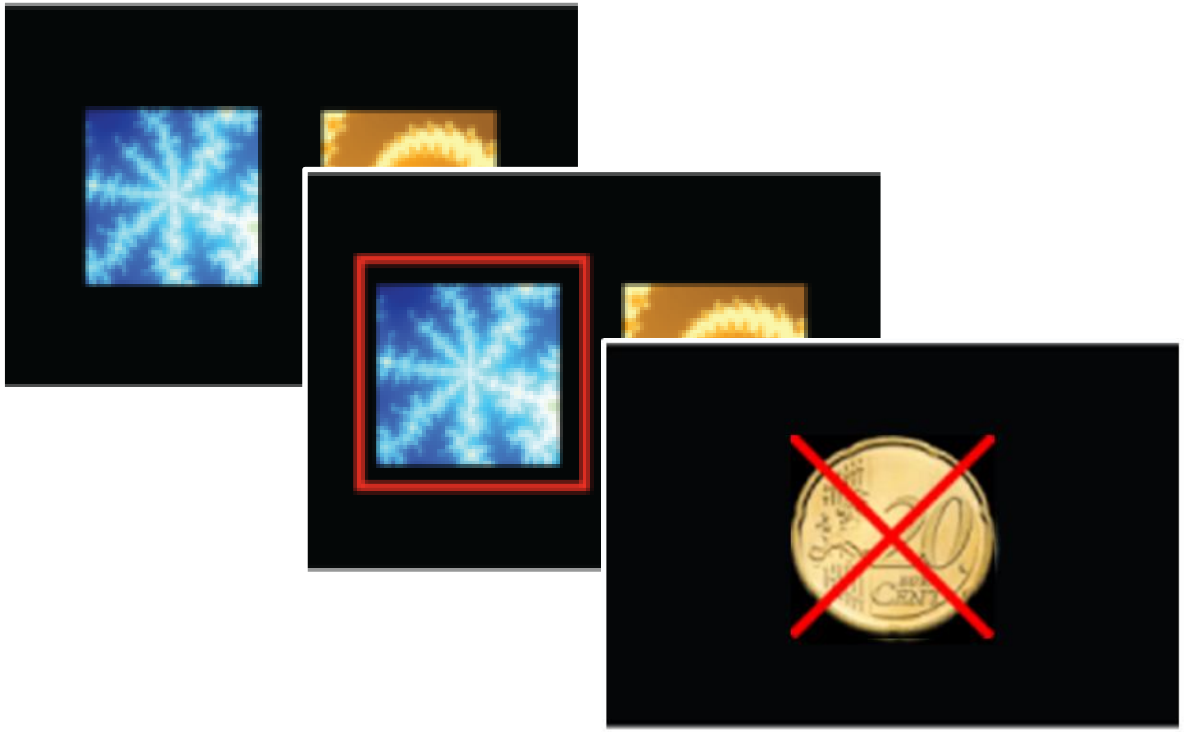
How is prediction shaped by learning?



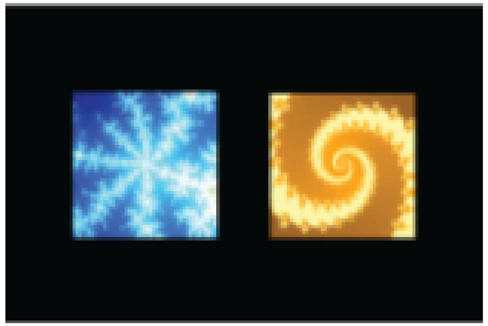


Sutton & Barto (1998); Rangel et al (2008); [Daw & Niv workshop 2015]

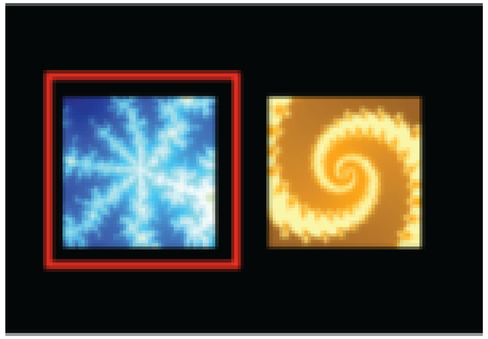
reward contingency 80:20



One simple experiment: two choice task



choice presentation



action selection

outcome

what do we know? — choice & outcome

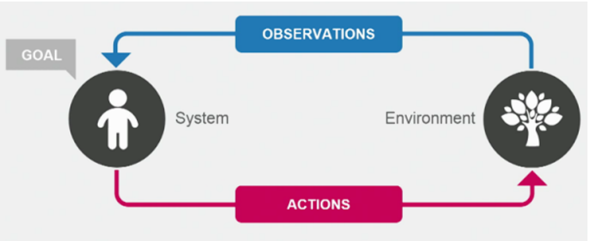
what can we measure? — choice accuracy

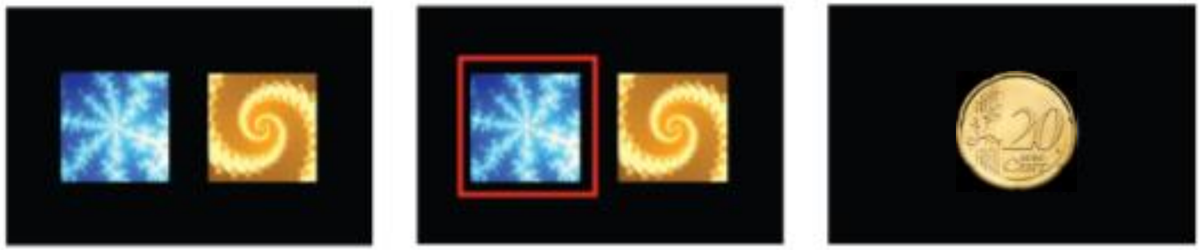
what do we not know? — RL update

	subjID	trialID	choice	outcome
1	1	1	1	1
2	1	2	1	1
3	1	3	1	1

p(choosing the better option)

Q-learning Value Update



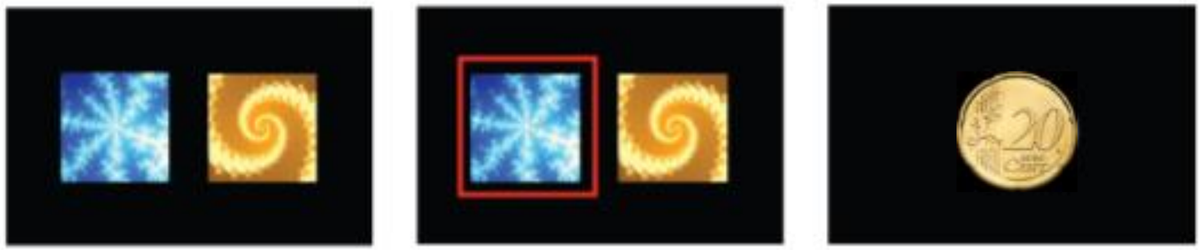


Cognitive Model
- cognitive process
- using internal variables and free parameters

Observation Model (Data Model)
- relate model to observed data
- has to account for noise

Q-Learning

- The idea: error-driven learning
- Change in value is proportional to the difference between actual and predicted outcome



Value update: $V_t = V_{t-1} + \alpha * \mathrm{PE}_{t-1}$

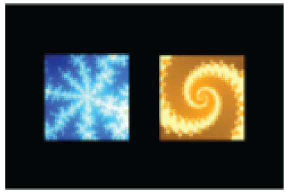
Prediction error: $\mathrm{PE}_{t-1} = R_{t-1} - V_{t-1}$

*Expectations on the next trial = the expectation on the current trial + learning rate * prediction error (reward – current expectation)*

Understand the learning rate

Value update: $V_t = V_{t-1} + \alpha * \mathrm{PE}_{t-1}$

Prediction error: $\mathrm{PE}_{t-1} = R_{t-1} - V_{t-1}$



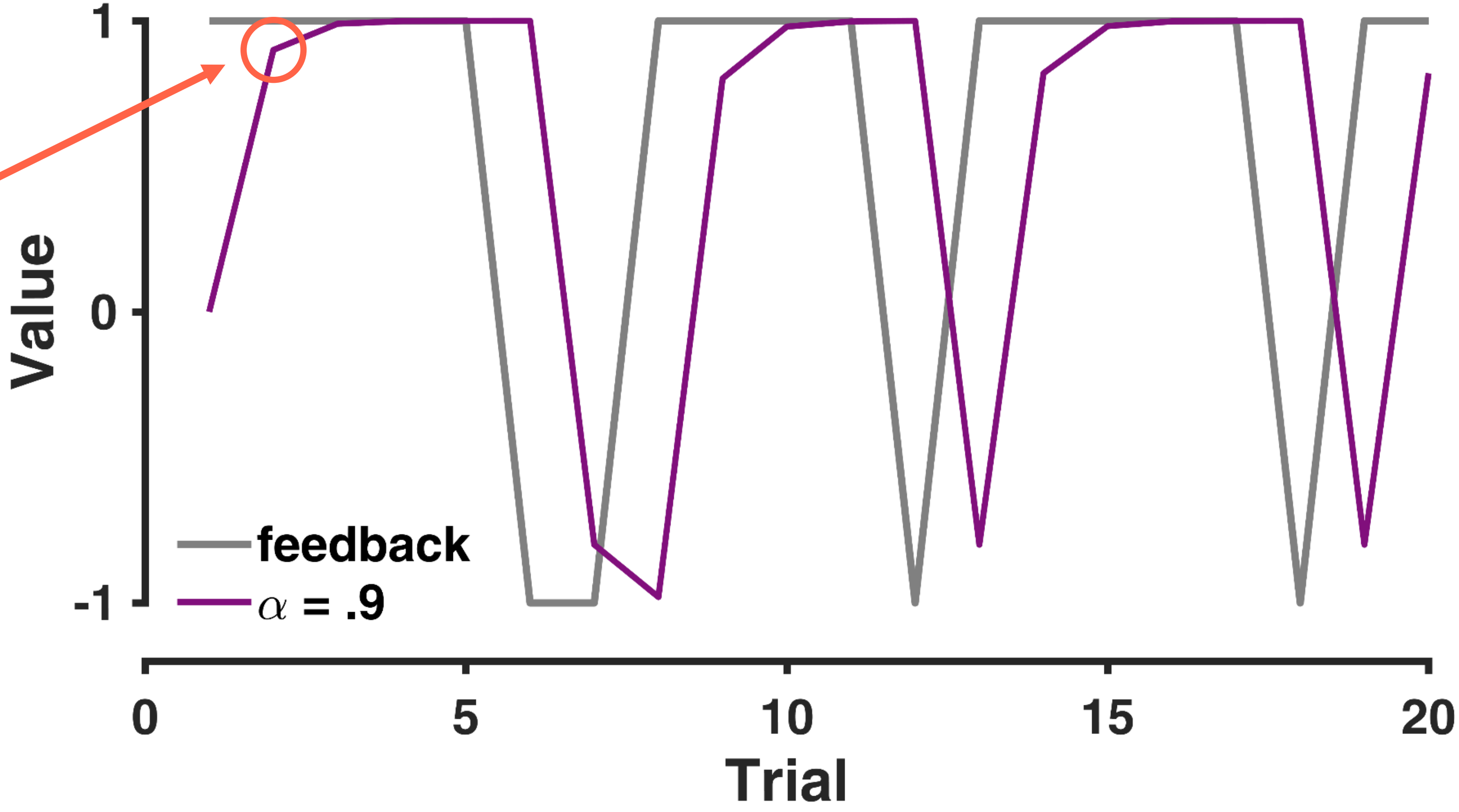
if $\alpha = 0.9$

$V_1 = 0$

$V_2 = V_1 + 0.9 * (1 - V_1)$

$= 0 + 0.9 * (1 - 0)$

$= 0.9$

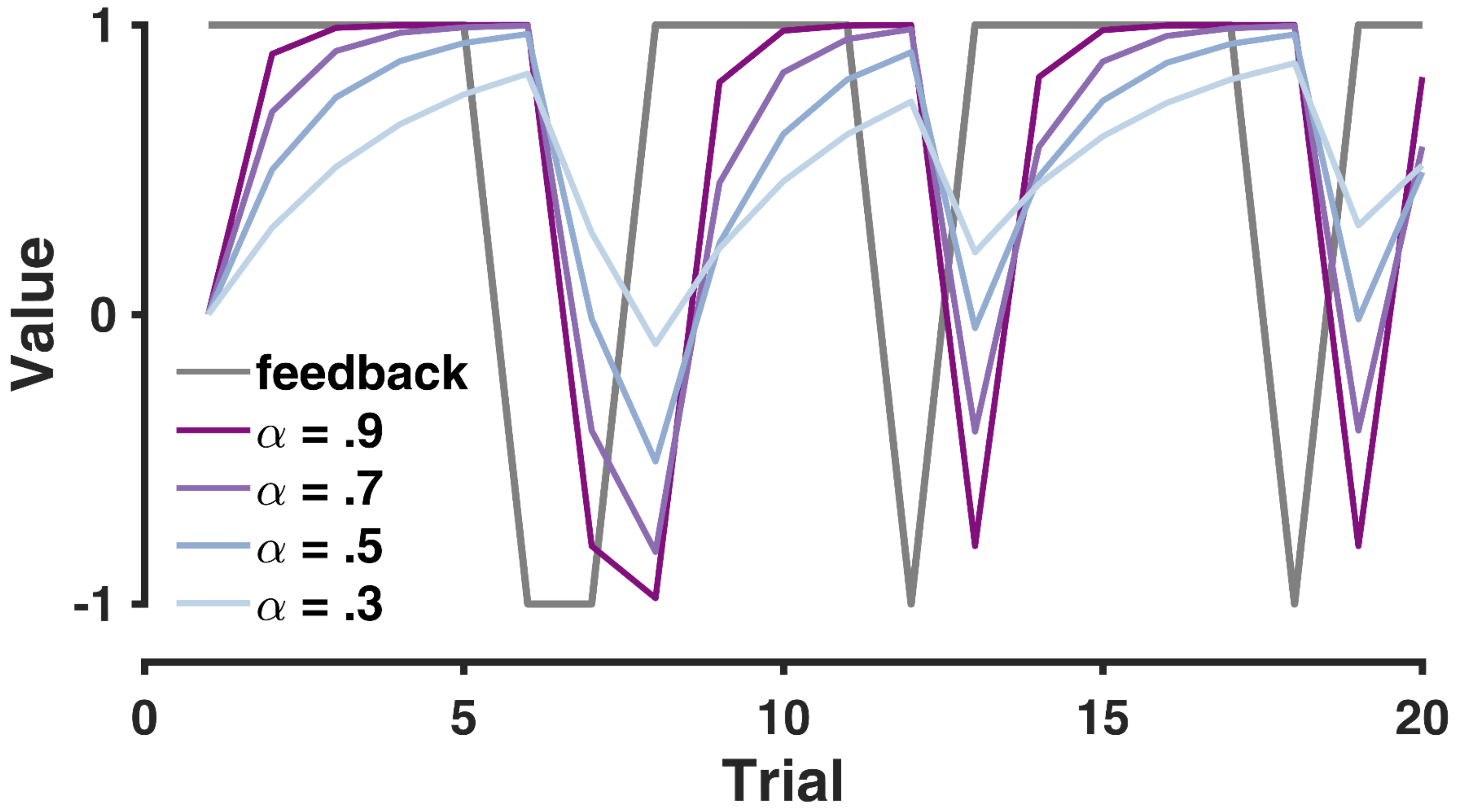


reward contingency – 80:20

Understand the learning rate

Value update: $V_t = V_{t-1} + \alpha * \mathrm{PE}_{t-1}$

Prediction error: $\mathrm{PE}_{t-1} = R_{t-1} - V_{t-1}$



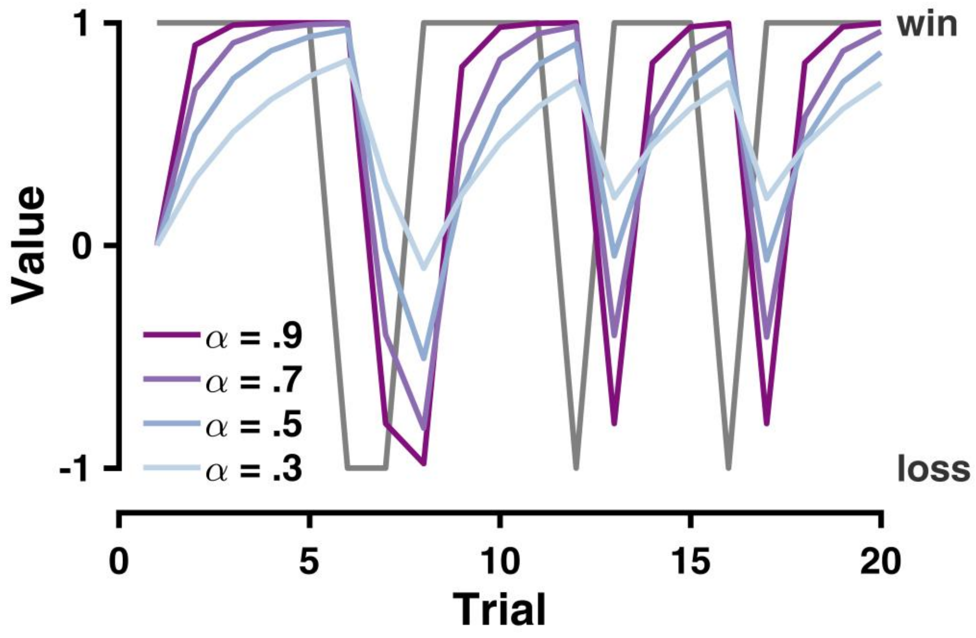
reward contingency – 80:20

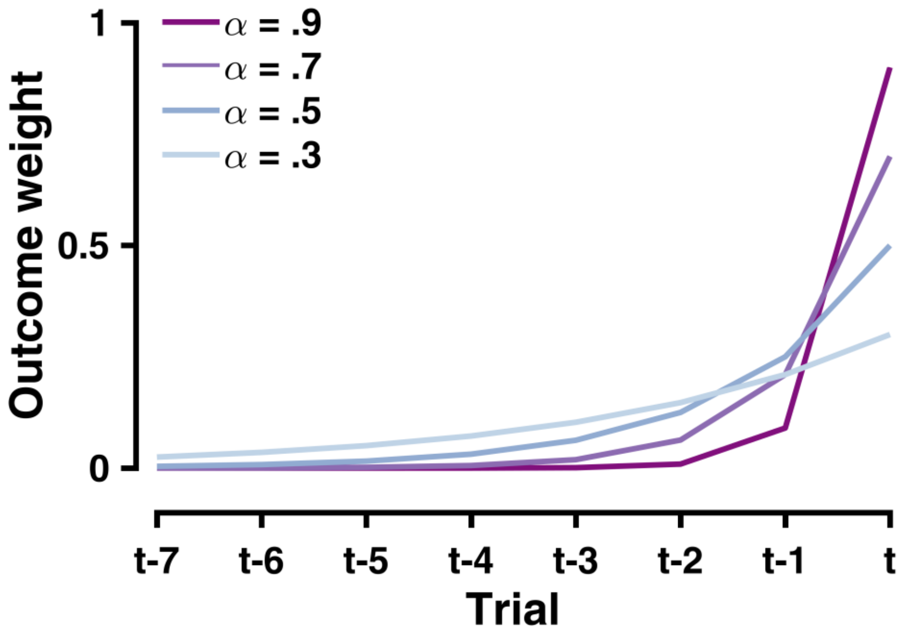
Understand the learning rate

Value update: $V_t = V_{t-1} + \alpha * \mathrm{PE}_{t-1}$

Prediction error: $\mathrm{PE}_{t-1} = R_{t-1} - V_{t-1}$

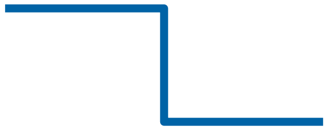
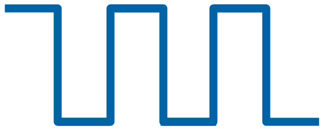
$$\begin{aligned} V_t &= (1-\alpha)\,V_{t-1} + \alpha R_{t-1} \\ &= (1-\alpha)\,(V_{t-2} + \alpha\,(R_{t-2} - V_{t-2})) + \alpha R_{t-1} \\ &= (1-\alpha)^t V_0 + \boxed{\sum_{i=1}^{t} (1-\alpha)^{t-i} \alpha R_i} \end{aligned}$$





Is there an optimal learning rate?

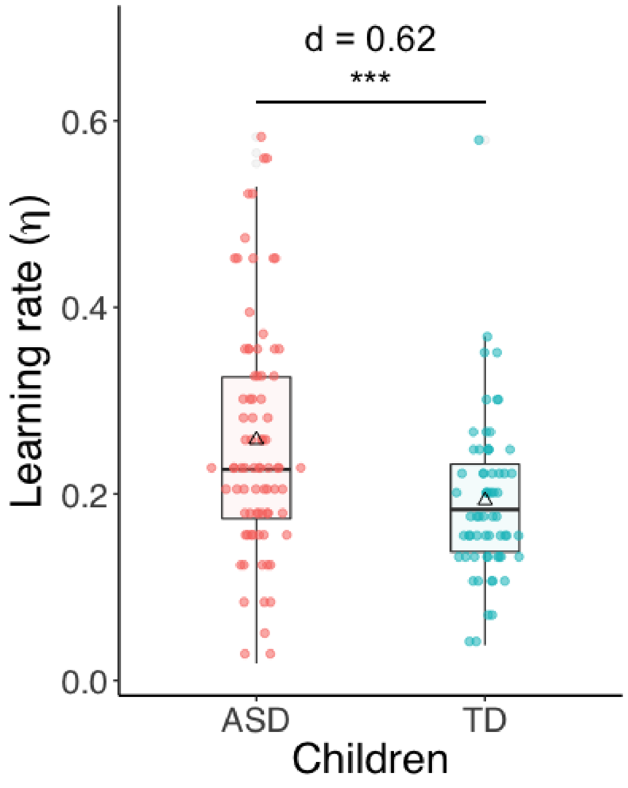
- it hugely depends on the task!

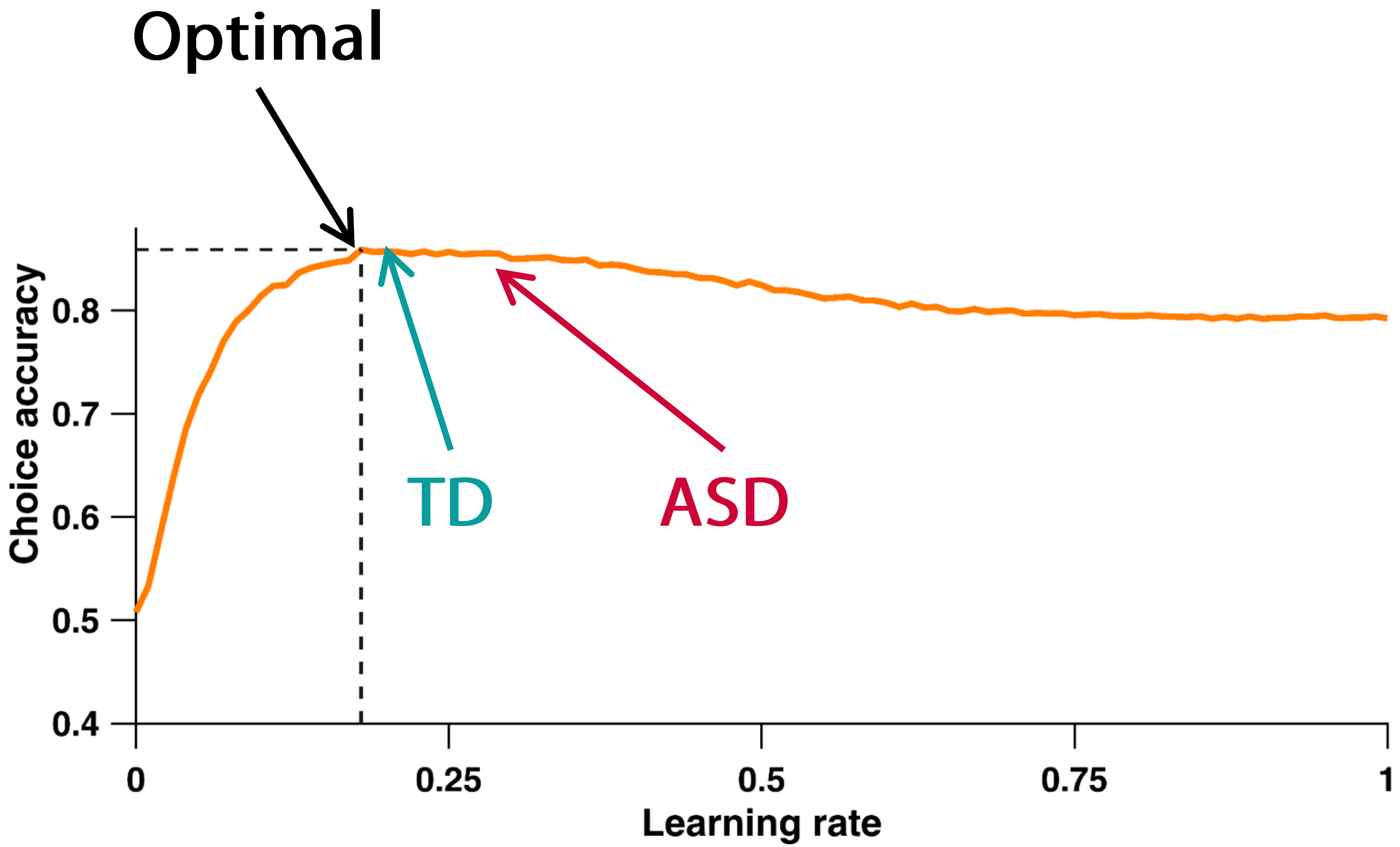
task structure			
reward schedule	**85:15**	**80:20**	**75:25**
number of trials	**40**	**80**	**100**

a task = task structure **+** reward schedule **+** number of trials

An optimal learning rate exists only when the task is settled.

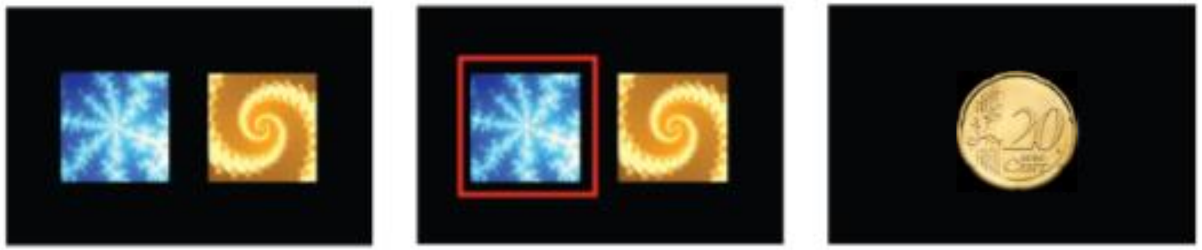
Optimal learning rate?





Crawley* & Zhang* et al., (2020)

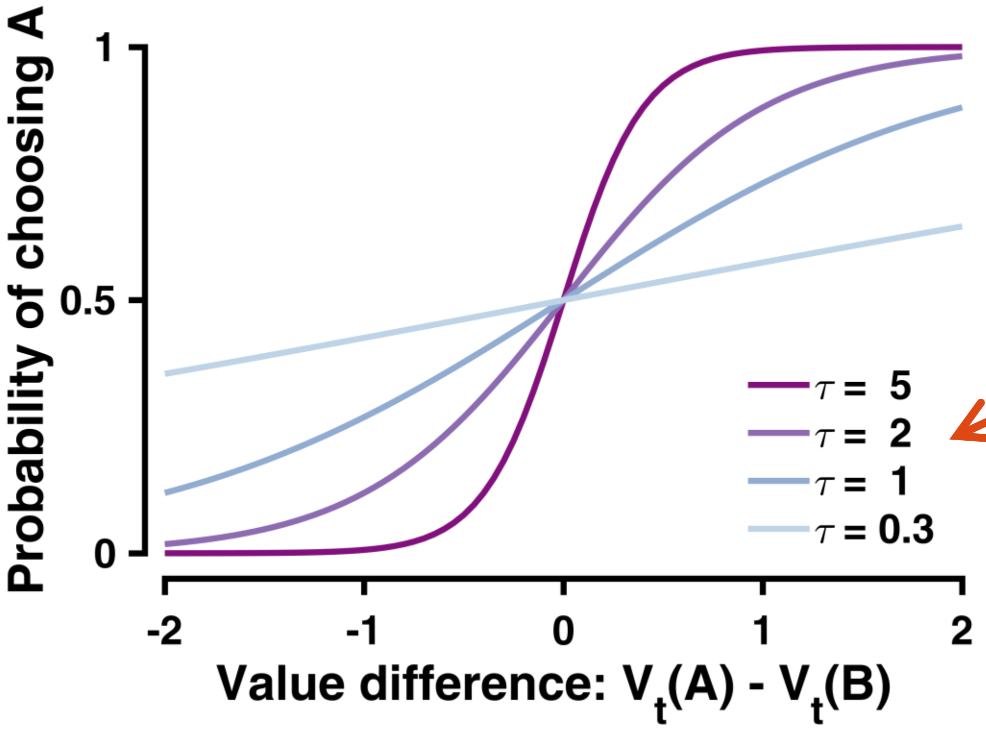
Choice rule: softmax

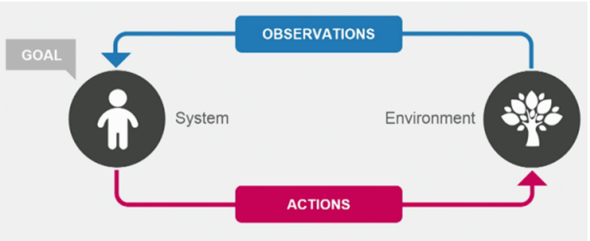


V()
V()

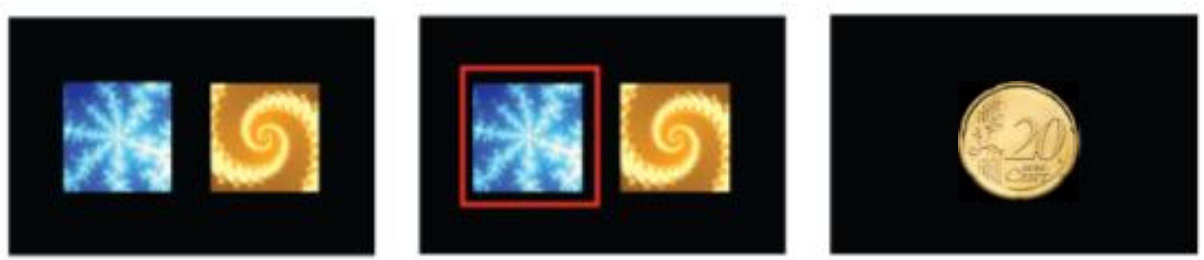
$$\boxed{p_t(A) = \frac{e^{\tau * V_t(A)}}{e^{\tau * V_t(A)} + e^{\tau * V_t(B)}}}$$

$$= \frac{1}{1 + e^{-\tau * (V_t(A) - V_t(B))}}$$





Q-Learning



Value update:

$$V_{t+1} = V_t + \alpha * PE_t$$

Prediction error:

$$PE_t = R_t - V_t$$

choice rule (sigmoid /softmax):

$$p(C{=}a) = \frac{1}{1+e^{\tau*(v(b)-v(a))}}$$

α - learning rate
PE - reward prediction error
V - value
R - reward
τ - softmax temperature

Some terminologies

simple Q-Learning

$$V_{t+1} = V_t + \alpha\,(R_t - V_t)$$

$$p_t(A) \;=\; \frac{1}{1+e^{-\tau*(\mathrm{v}(A)-\mathrm{v}(B))}}$$

$$\mathrm{LL} = \mathrm{categorical\_lpmf}(\,\mathrm{C}_t \mid p_t)$$

simple linear regression

$$\mu = \mathrm{a} + \mathrm{b}\,\mathrm{X}$$

$$\mathrm{LL} = \mathrm{normal\_lpdf}(\,\mathrm{Y} \mid \mu,\, \delta\,)$$

Some terminologies

simple Q-Learning

$$V_{t+1} = V_t + \alpha\,(R_t - V_t)$$

$$p_t(\mathrm{A}) \;=\; \frac{1}{1+e^{-\tau*(\mathrm{v}(A)-\mathrm{v}(B))}}$$

$$\mathrm{LL} = \mathrm{categorical\_lpmf}(\, \mathrm{C}_t \mid p_t)$$

simple linear regression

$$\mu = \mathrm{a} + \mathrm{b}\;\mathrm{X}$$

$$\mathrm{LL} = \mathrm{normal\_lpdf}(\, \mathrm{Y} \mid \mu,\, \delta\,)$$

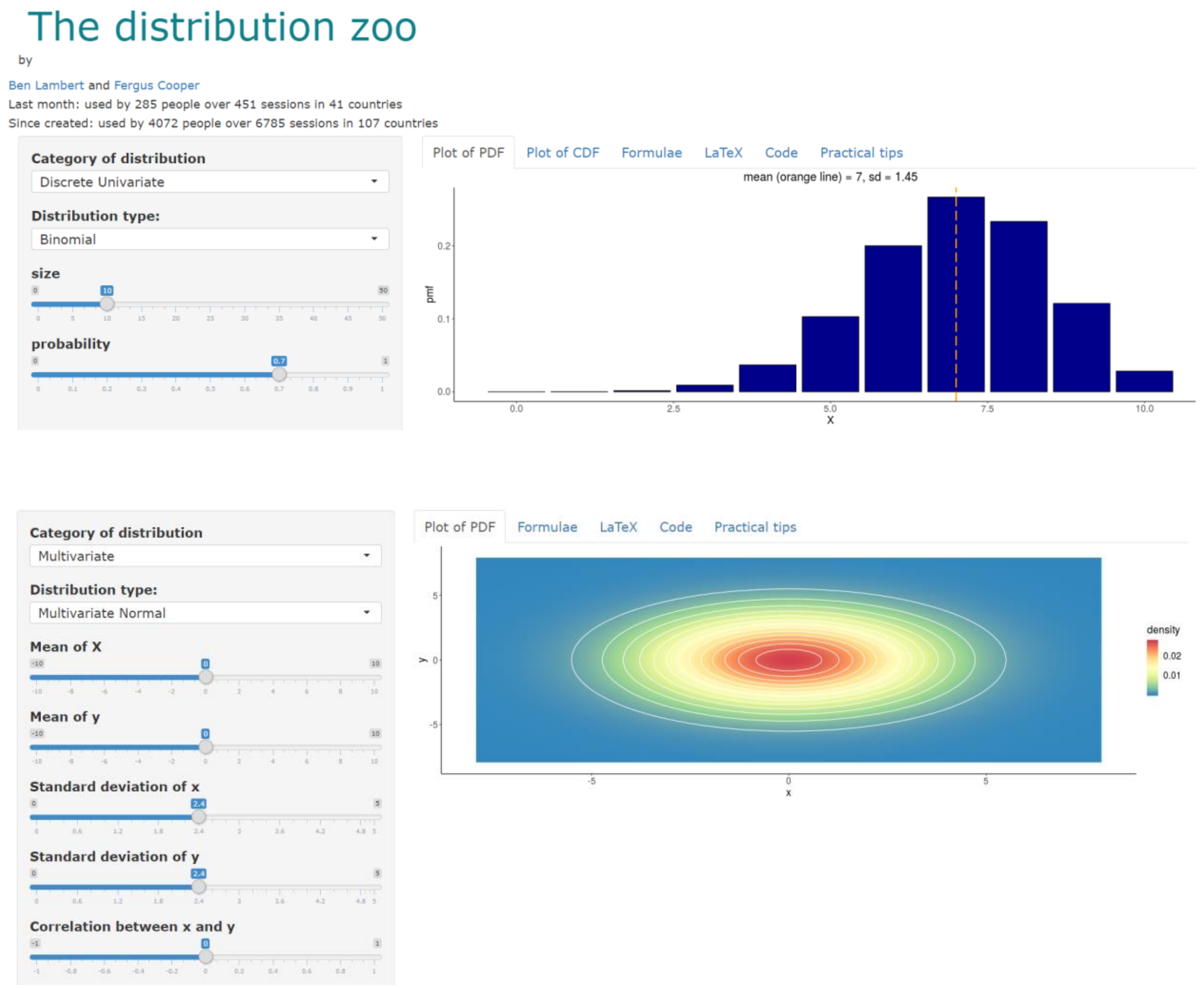
data / observation

free parameter / model parameter

internal/latent (model) variable

likelihood function

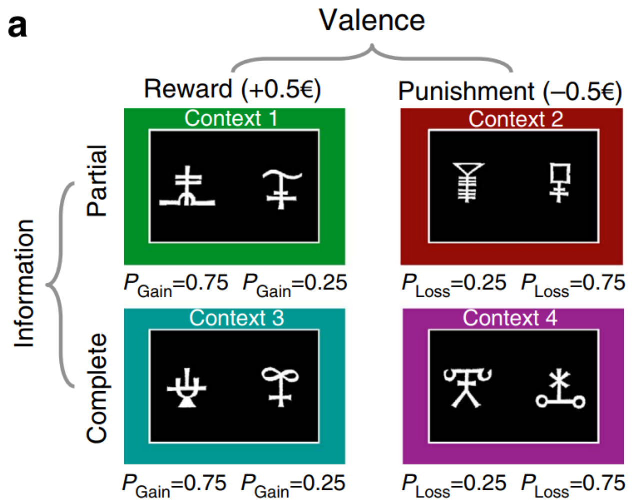
How do I know which likelihood to use?



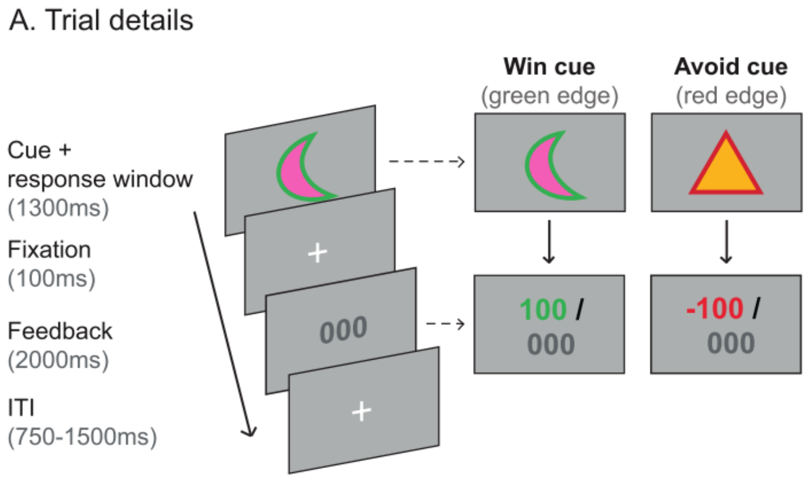
Lambert (2018)

https://ben18785.shinyapps.io/distribution-zoo/

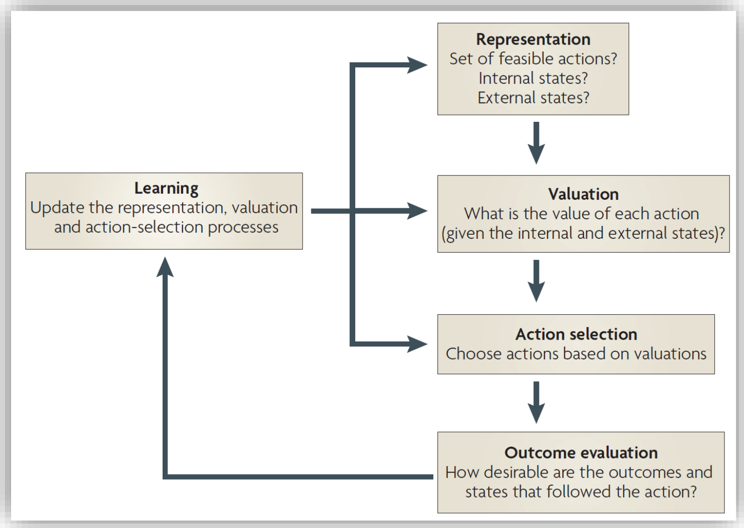
Generalizing RL framework

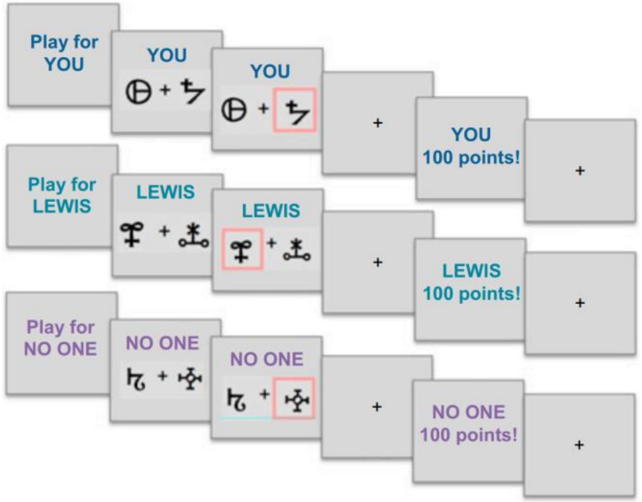


Palminteri et al. (2015)

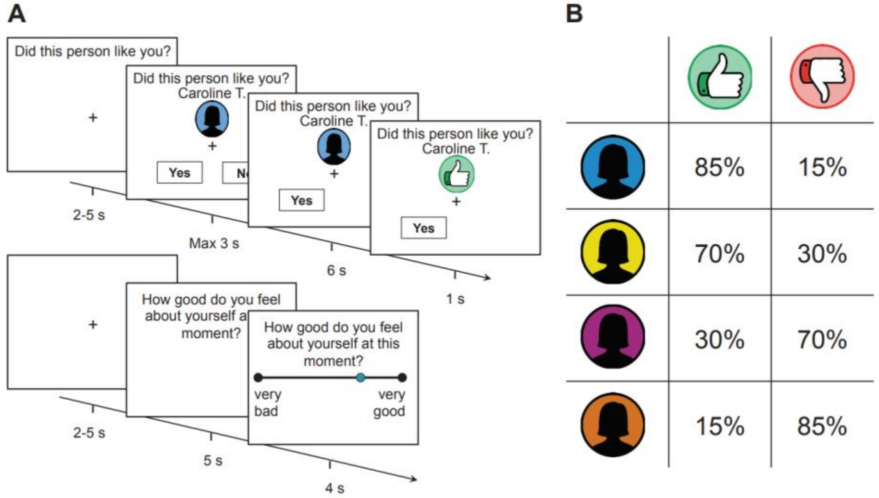


Swart et al. (2017)





Lockwood et al. (2016)

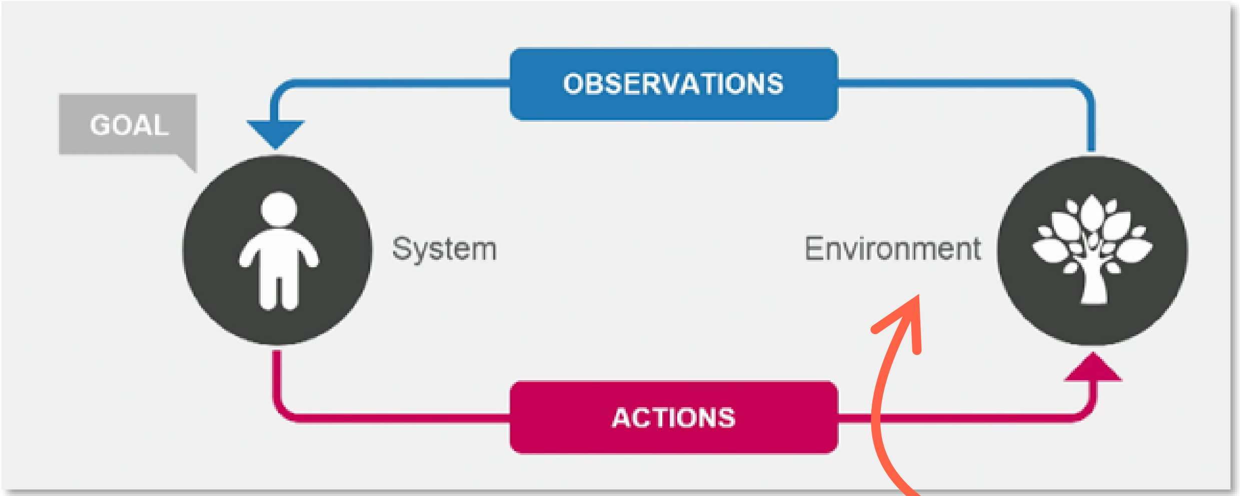


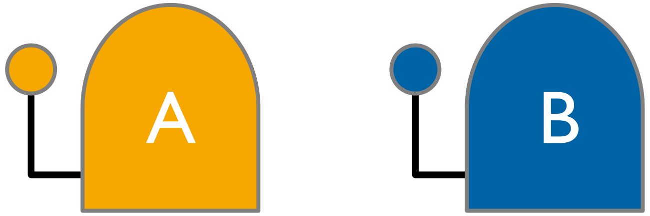
Will et al. (2017)

Social learning & decision-making

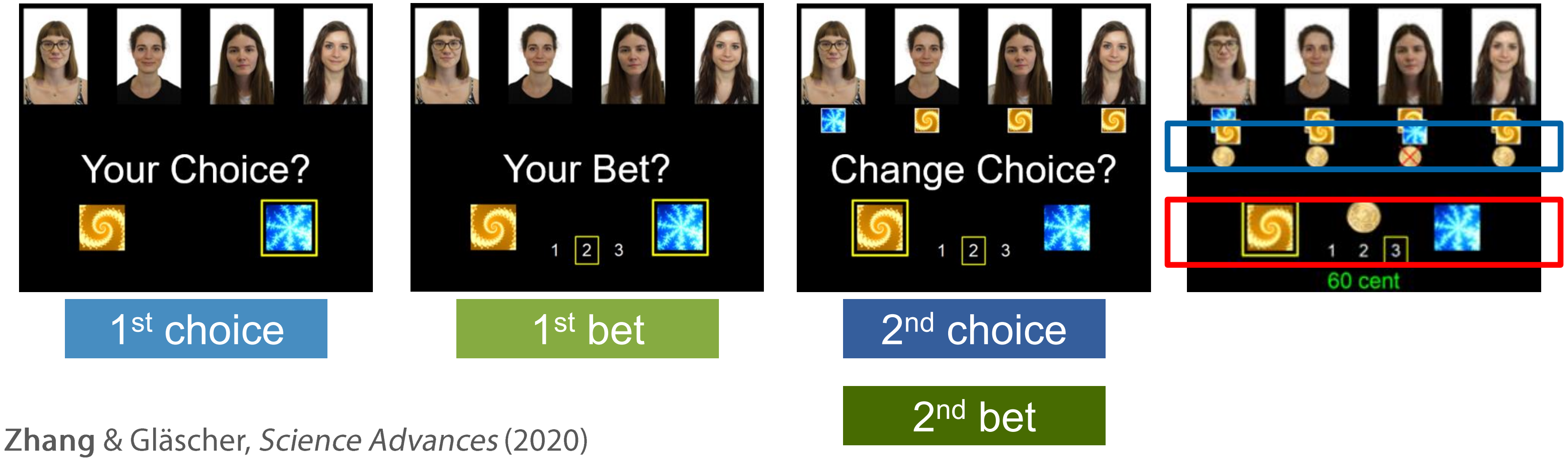
Decision-making stage (value type)	Target (and reference frame) of valuation		
	Value of other people (for oneself)	Value of other people's experiences (for them)	Value of social constellation (for normative social principle)
Choice (decision value and anticipated value)	Deciding whether to marry someone	Choosing a school for your child	Altruistic punishment of norm violations
Outcome (experienced value)	Being smiled at	Empathy with someone's pain	Enjoying fair distributions
Learning (anticipated value)	Learning about someone's trustworthiness	Learning about someone else's preferences	Changing your opinion to increase social conformity

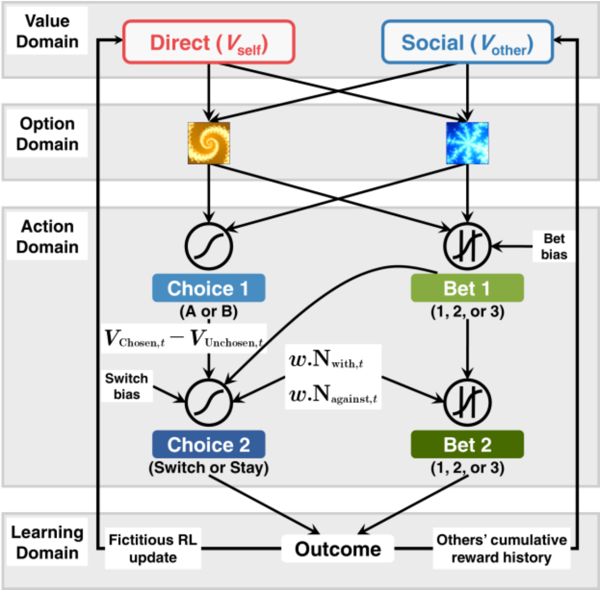
learn from others (to benefit oneself)





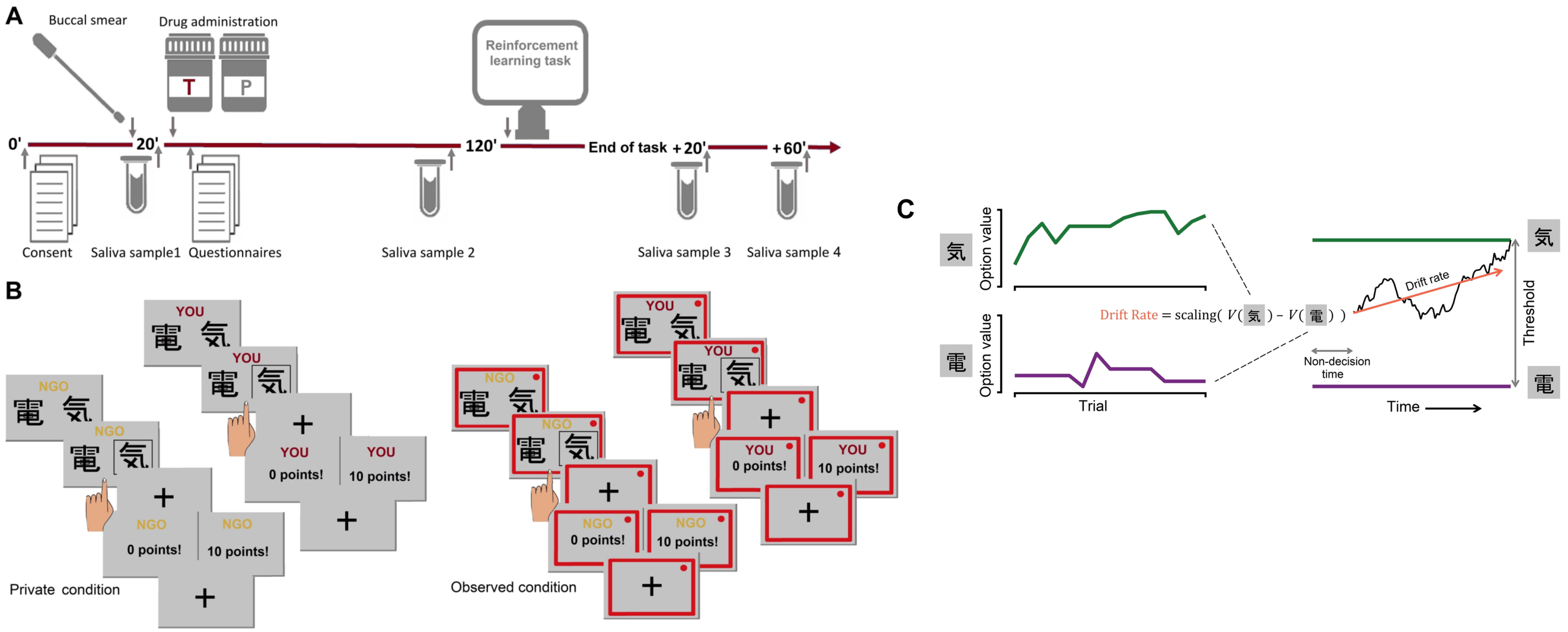
Social environment





Zhang & Gläscher, *Science Advances* (2020)

learn for others (pro-social)



Kutlikova* & **Zhang***, et al., *Neuropsychopharmacology* (2023)

Some more food for thought

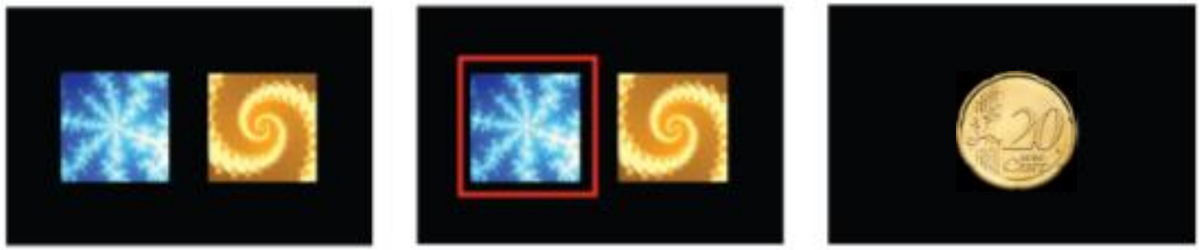
What is a social RL task?	There are commonly two variations: either reward learning in social contexts (e.g. learn to expect monetary reward for a social partner) or social feedback learning (e.g. learn to expect social status or social evaluation). When the goal is to compare different types of feedback (e.g. social vs. non-social feedback), we suggest matching the feedback as closely as possible on 'domain general' properties, such as salience or preference	Lockwood *et al.*, 2016; Will *et al.*, 2017
Is the learning rate static across trials, or dynamically adapting along the course of the experiment?	The learning rate does not necessarily have to be constant. But in the case of the Rescorla–Wagner model (and related models), the learning rate is indeed static. A dynamic learning rate, however, is possible when other types of models are applied. Note that the interpretation of the learning rate we discussed in the main text is independent of this constant *vs* dynamic property	Li *et al.*, 2011; Mathys *et al.*, 2011
Is it possible to use RL models in the absence of choice data?	At least some sort of data is needed to perform model estimation. For example, skin conductance response (SCR) or pupil size response (PSR) have been used to fit RL models in associative fear learning tasks, where choice data was not available	Li *et al.*, 2011; Tzovara *et al.*, 2018

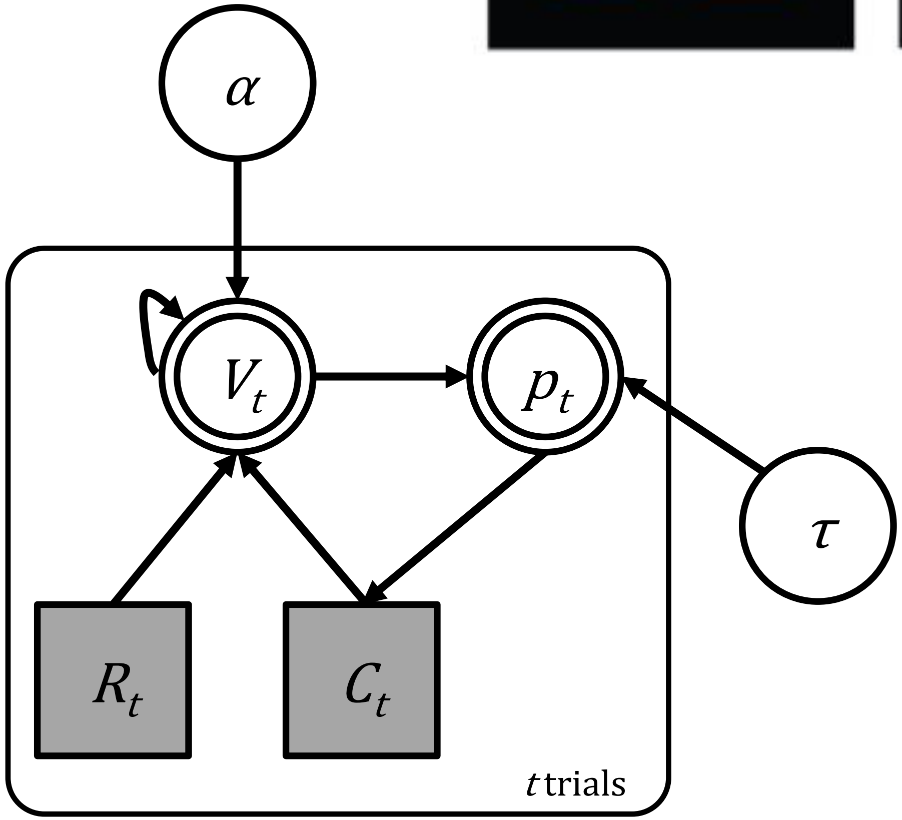
Zhang et al., (2020)

Outline

- Recap on Bayesian modeling
- Why Stan?
- What is (computational) cognitive modeling?
- The idea of the simple Q-Learning model
- Implementing Q-Learning model for one subject in Stan
- Fitting multiple subjects with the hBayesDM package
- Q&A

RL – Implementation





$$\alpha \sim Uniform(0,1)$$

$$\tau \sim Uniform(0,3)$$

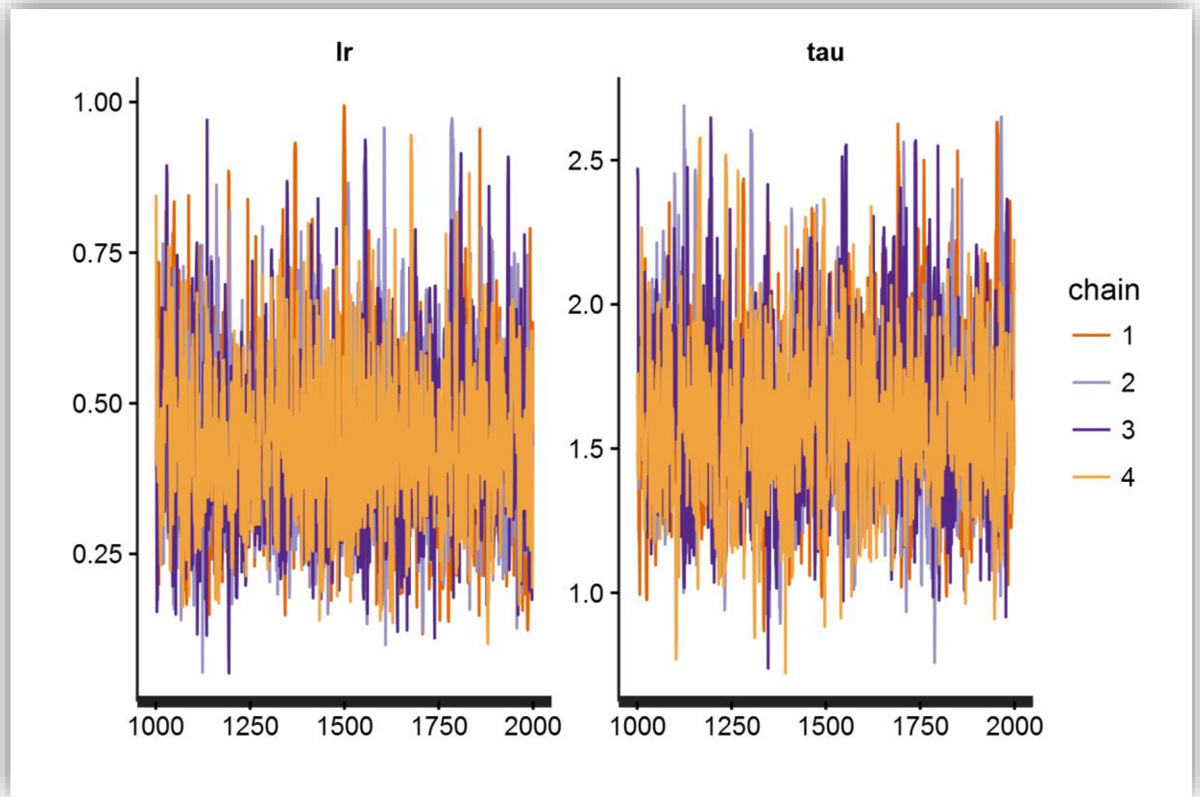
$$p_t(C = A) = \frac{1}{1 + e^{\tau(V_t(B) - V_t(A))}}$$

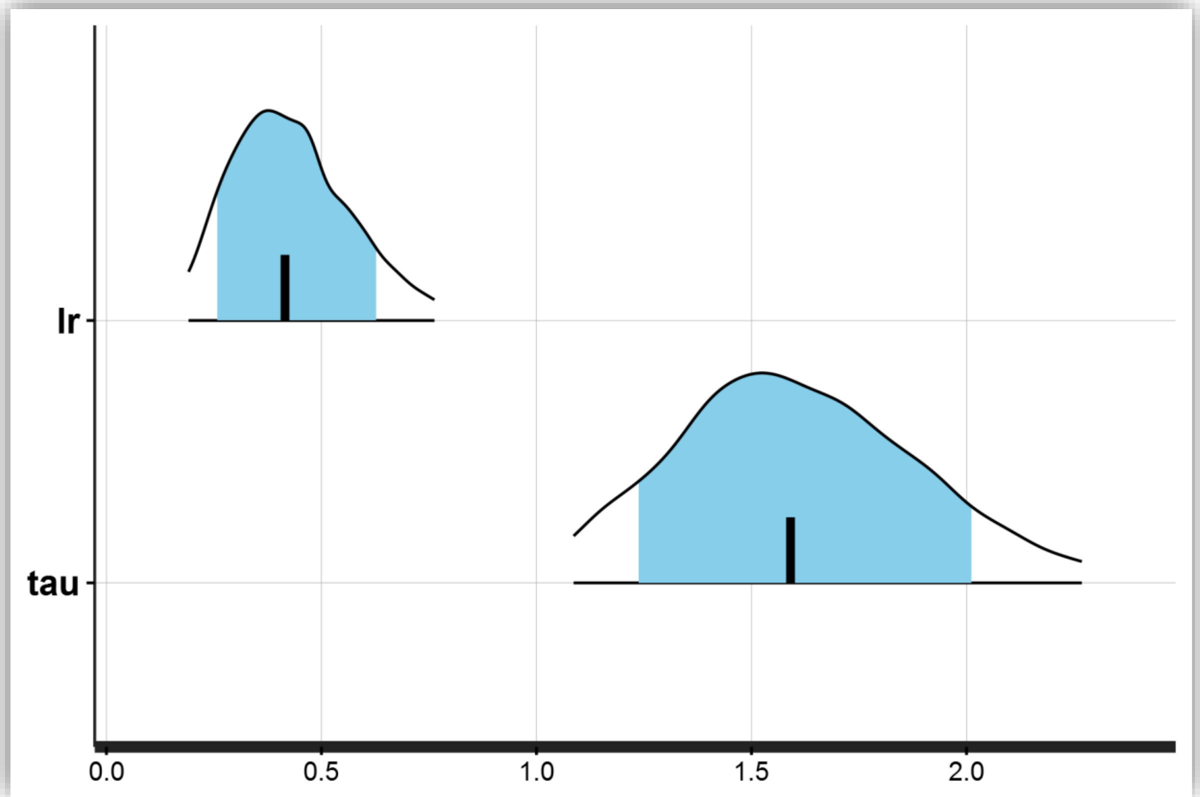
$$V_{t+1}^{c} = V_t^{C} + \alpha\left(R_t - V_t^{C}\right)$$

deterministic variable

Live demo of building the Q-Learning model model in Stan from scratch

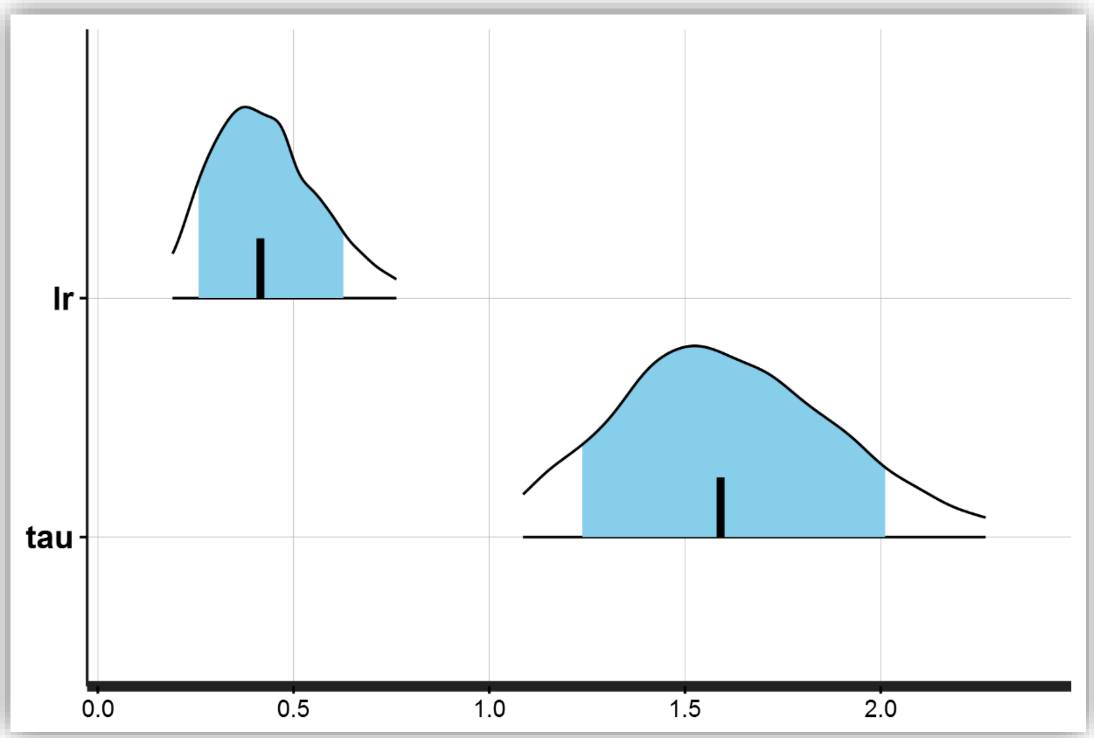
RL – MCMC Output



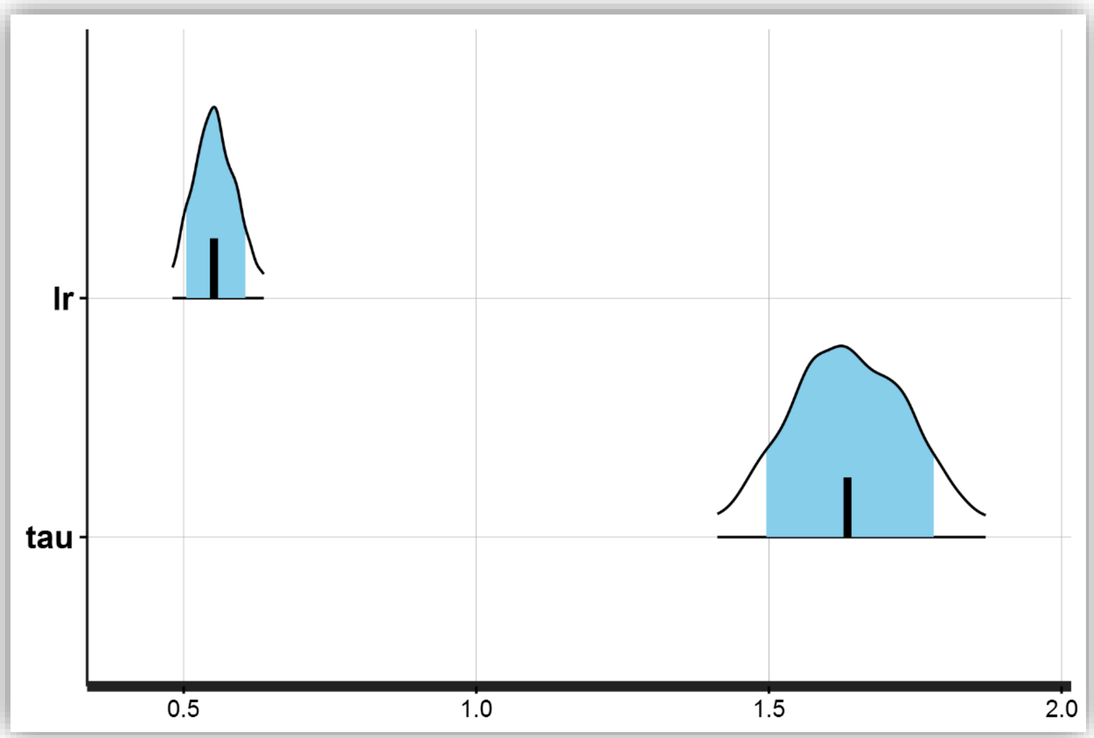


What if we have more data?

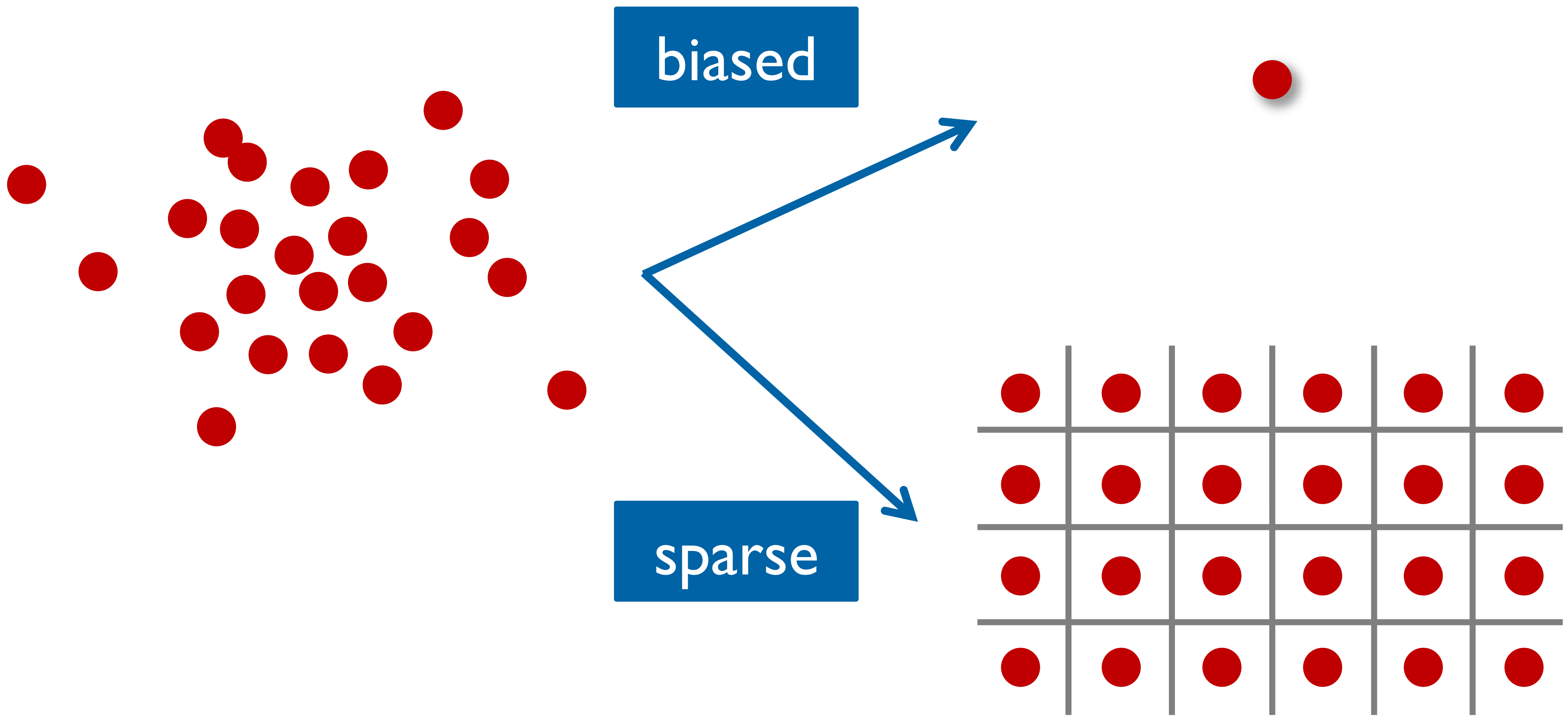
N = 1



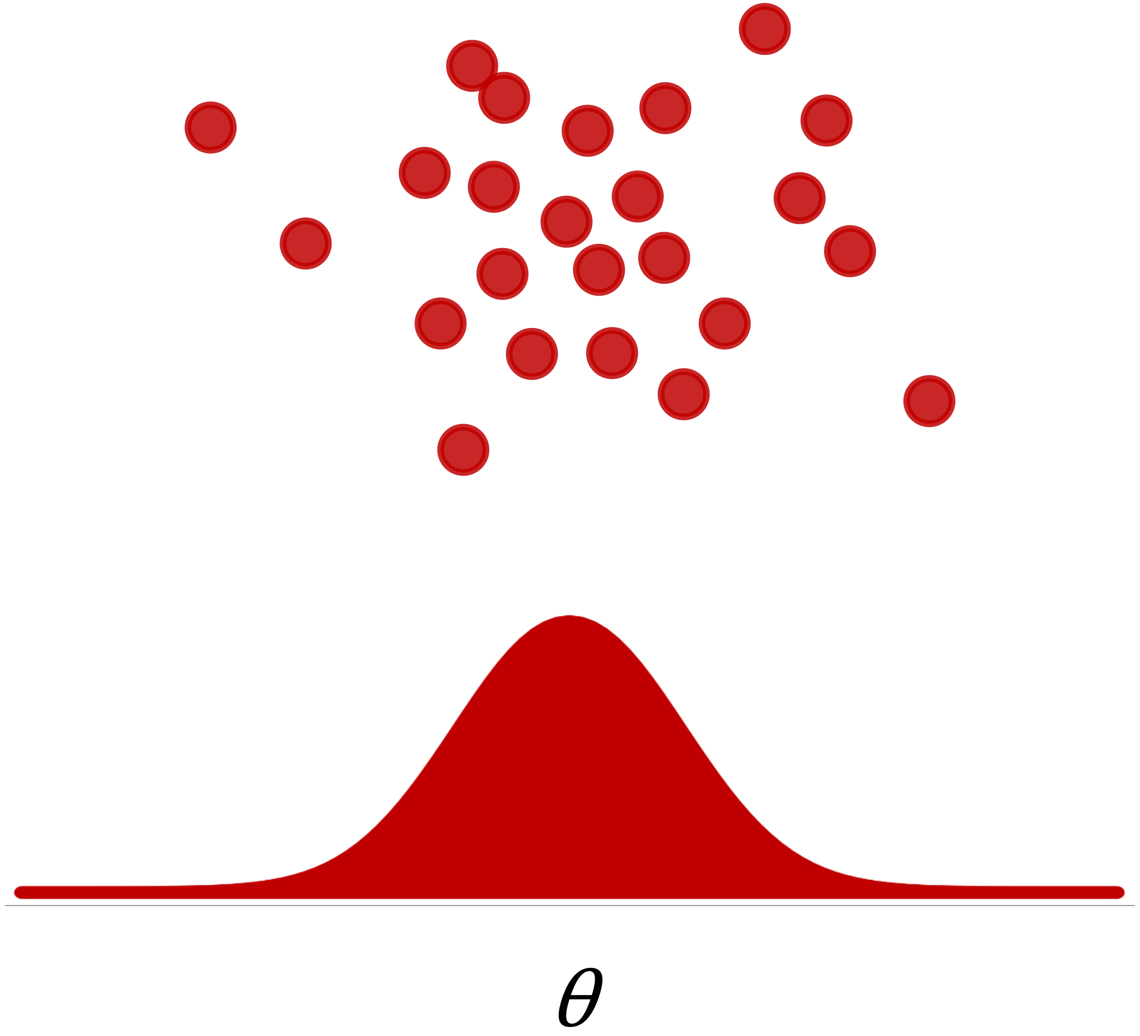
N = 10

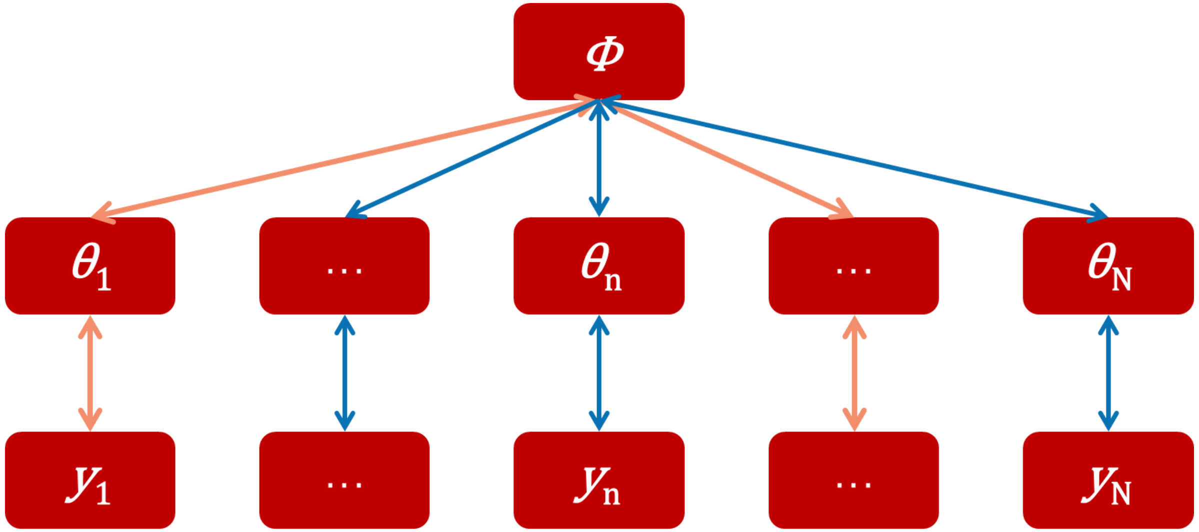


Fitting Multiple Participants

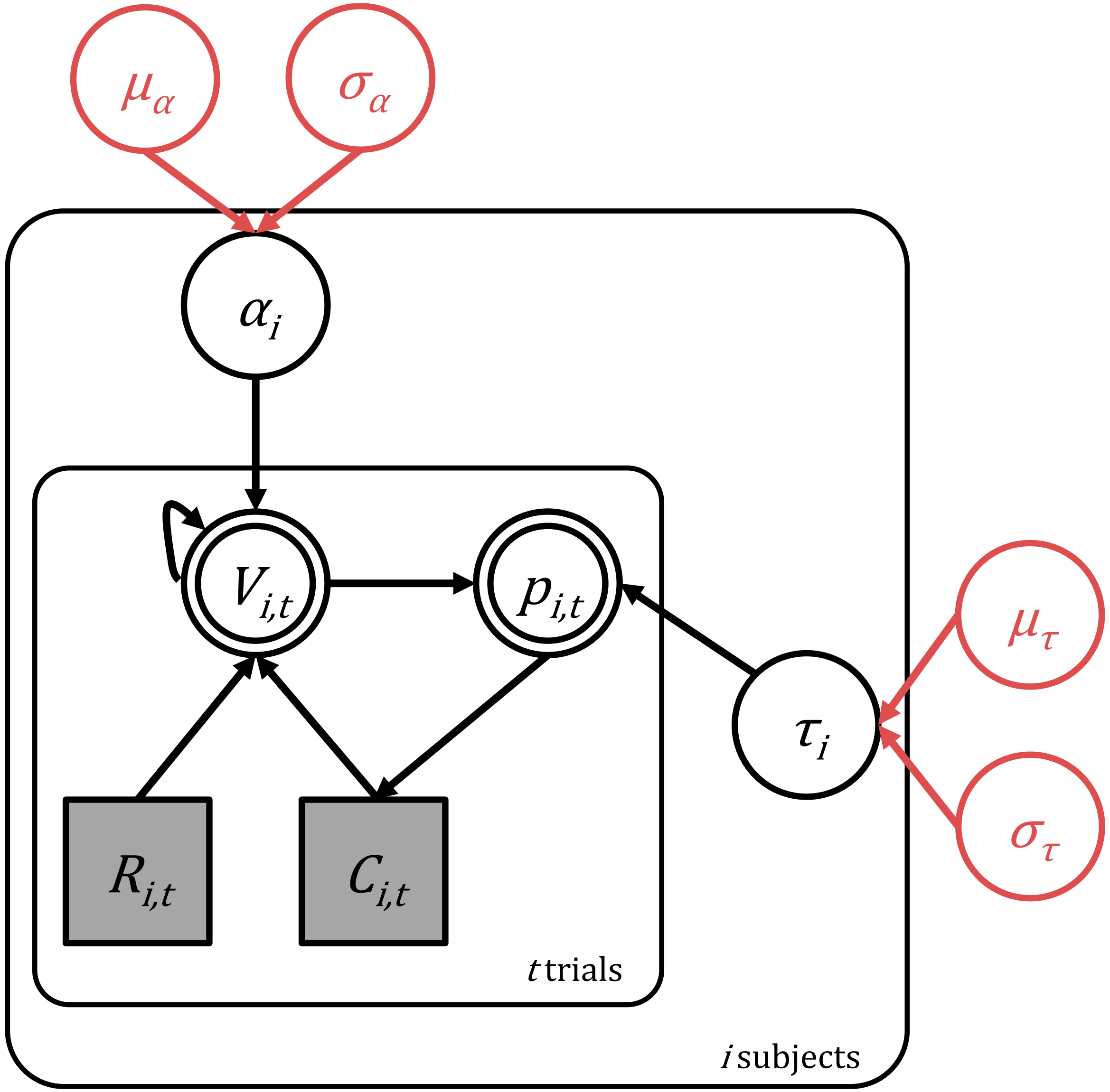


Fitting Multiple Participants with hierarchical Bayesian analysis (HBA)





Hierarchical RL Model



Information Criteria

AIC – Akaike information criterion

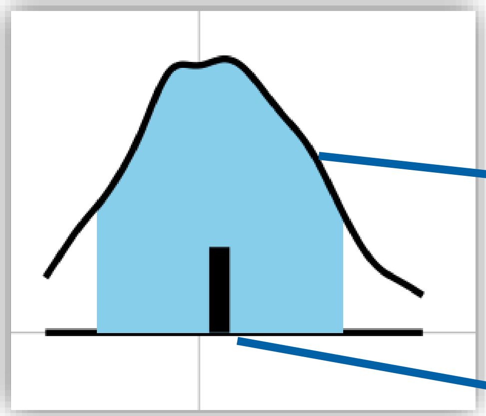
DIC – Deviance Information Criterion

WAIC – Widely Applicable Information Criterion

(Watanabe–Akaike information criterion)

finding the model that has the highest out-of-sample predictive accuracy

approximation to LOO



WAIC: using entire posterior distribution

AIC/DIC: using point estimation

Compute WAIC from Likelihood

$$\text{WAIC} = -2\,\widehat{\text{elpd}}_{\text{waic}}$$

expected log pointwise predictive density

$$\widehat{\text{elpd}}_{\text{waic}} = \widehat{\text{lpd}} - \widehat{p}_{\text{waic}}$$

$$\begin{aligned} \widehat{\text{lpd}} &= \text{computed log pointwise predictive density} \\ &= \sum_{i=1}^{n} \log\left(\frac{1}{S}\sum_{s=1}^{S} p(y_i|\theta^s)\right). \end{aligned}$$

estimated effective number of parameters

$$\widehat{p}_{\text{waic}} = \sum_{i=1}^{n} V_{s=1}^{S}\left(\log p(y_i|\theta^s)\right)$$

```
lpd <- log(colMeans(exp(log_lik)))
```

```
p_waic <- colVars(log_lik)
```

Vehtari & Gelman (2014)

The {loo} Package

```
> library(loo)
> LL1   <- extract_log_lik(stanfit)
> loo1  <- loo(LL1)    # PSIS leave-one-out
> waic1 <- waic(LL1)   # WAIC
```

Pareto Smoothed Importance Sampling

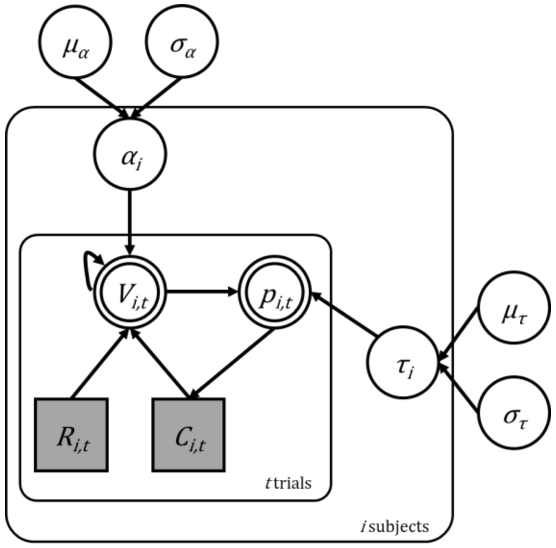
```
Computed from 4000 by 20 log-likelihood matrix

          Estimate  SE
elpd_loo     -29.5 3.3
p_loo          2.7 1.0
looic         58.9 6.7
```

Vehtari et al. (2015)

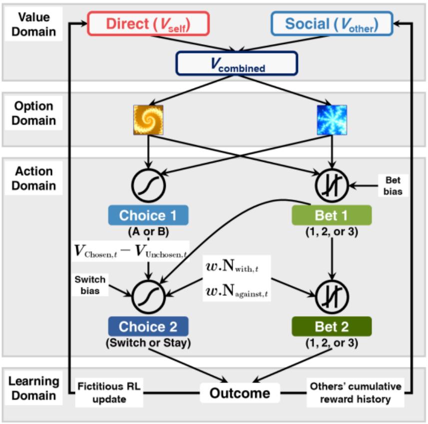
Recording the Log-Likelihood in Stan

```
generated quantities {
  ...
  real log_lik[nSubjects];
  ...

  { # local section, this saves time and space
    for (s in 1:nSubjects) {
      vector[2] v;
      real pe;

      log_lik[s] = 0;
      v = initV;

      for (t in 1:nTrials) {
        log_lik[s] = log_lik[s] + categorical_logit_lpmf(choice[s,t] | tau[s] * v);

        pe = reward[s,t] - v[choice[s,t]];
        v[choice[s,t]] = v[choice[s,t]] + lr[s] * pe;
```

HBA sounds good, but...

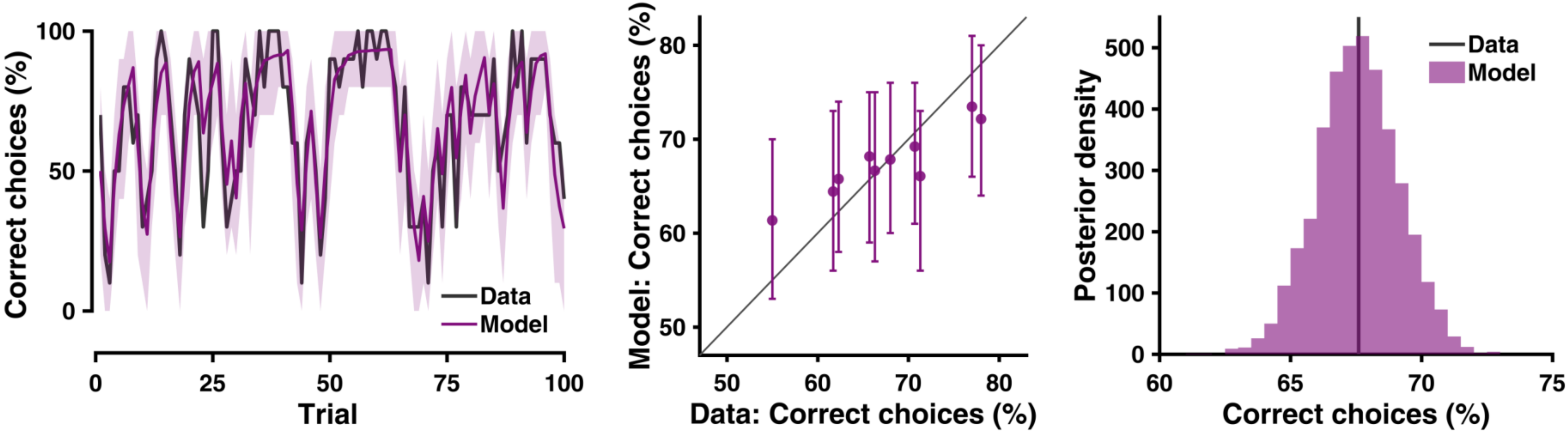


$$\mu_\alpha \sim Uniform(0,1)$$
$$\sigma_\alpha \sim halfCauchy(0,1)$$
$$\mu_\tau \sim Uniform(0,3)$$
$$\sigma_\tau \sim halfCauchy(0,3)$$
$$\alpha_i \sim Normal(\mu_\alpha, \sigma_\alpha)_{\mathcal{T}(0,1)}$$
$$\tau_i \sim Normal(\mu_\tau, \sigma_\tau)_{\mathcal{T}(0,3)}$$
$$p_{i,t}(C=A) = \frac{1}{1+e^{\tau_i(V_{i,t}(B)-V_{i,t}(A))}}$$
$$V^c_{i,t+1} = V^C_{i,t} + \alpha_i(R_{i,t} - V^C_{i,t})$$



$$\begin{aligned}
\mathbb{V}_{self,t} &= [V_{self,t}(A), V_{self,t}(B)] \\
\mathbb{V}_{other,t} &= [V_{other,t}(A), V_{other,t}(B)] \\
\mathbb{V}_t &= \beta_{vself}\,\mathbb{V}_{self,t} + \beta_{vother}\,\mathbb{V}_{other,t} \\
C1_t &\sim Categorical(Softmax(\mathbb{V}_t)) \\
U_{bet1,t} &= \beta_{bias_{c1}} + \beta_{vdiff_{c1}}(V_{chosen,C1,t} - V_{unchosen,C1,t}) \\
B1_t &\sim OrderedLogistic(U_{bet1,t} \mid \theta) \\
w.N_{against,t} &= \frac{\sum_{s=1}^{K} w_{s,t}}{\sum_{s=1}^{4} w_{s,t}}, K = 0,1,\ldots,4 \\
w.N_{with,t} &= \frac{\sum_{s=1}^{4-K} w_{s,t}}{\sum_{s=1}^{4} w_{s,t}} \\
V_t(\text{switch}) &= \beta_{bias_{c2}} + \beta_{vdiff_{c2}}(V_{chosen,C1,t} - V_{unchosen,C1,t}) + \beta_{against}\,w.N_{against,t} \\
C2 &\sim Bernoulli(V_t(\text{switch})) \\
U_{bet2,t} &= \begin{cases} U_{bet1,t} + \beta_{with_{agr}}\,w.N_{with,t} + \beta_{against_{agr}}\,w.N_{against,t} & \text{, if } C1 = C2 \\ U_{bet1,t} + \beta_{with_{dis}}\,w.N_{with,t} + \beta_{against_{dis}}\,w.N_{against,t}, & \text{if } C1 \neq C2 \end{cases} \\
B2_t &\sim OrderedLogistic(U_{bet2,t} \mid \theta) \\
\Phi(x) &= \frac{1}{1+e^{-x}} \\
\delta_{self,chosen,C2,t} &= R_{self,t} - V_{self,chosen,C2,t} \\
\delta_{self,unchosen,C2,t} &= -R_{self,t} - V_{self,unchosen,C2,t} \\
V_{self,chosen,C2,t+1} &= V_{self,chosen,C2,t} + \alpha\delta_{self,chosen,C2,t} \\
V_{self,unchosen,C2,t+1} &= V_{self,unchosen,C2,t} + \alpha\delta_{self,unchosen,C2,t}
\end{aligned}$$

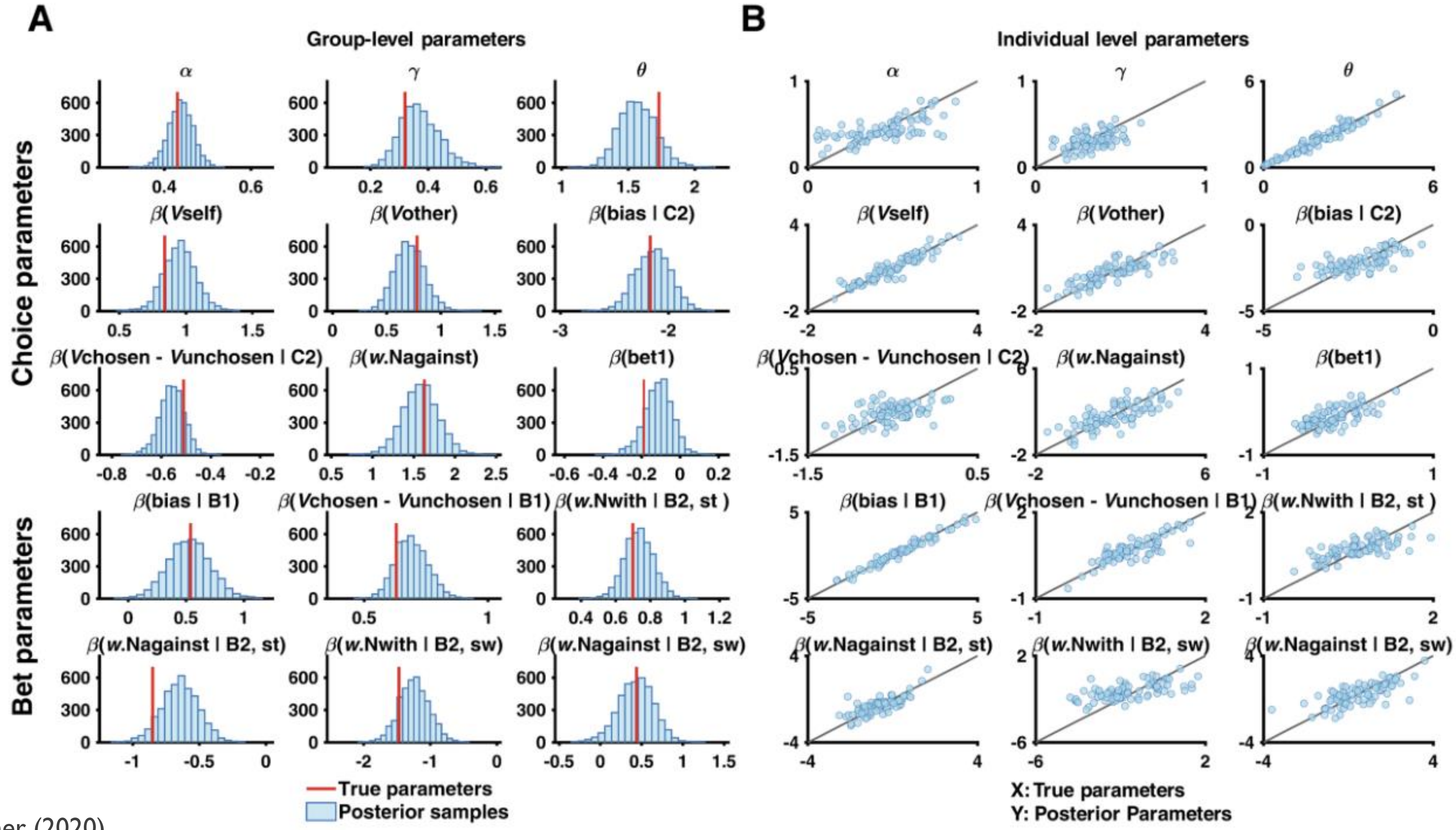
Posterior predictive check:
can the model resonate with data?



Zhang et al. (2020)

Parameter recovery:

are parameters identifiable?

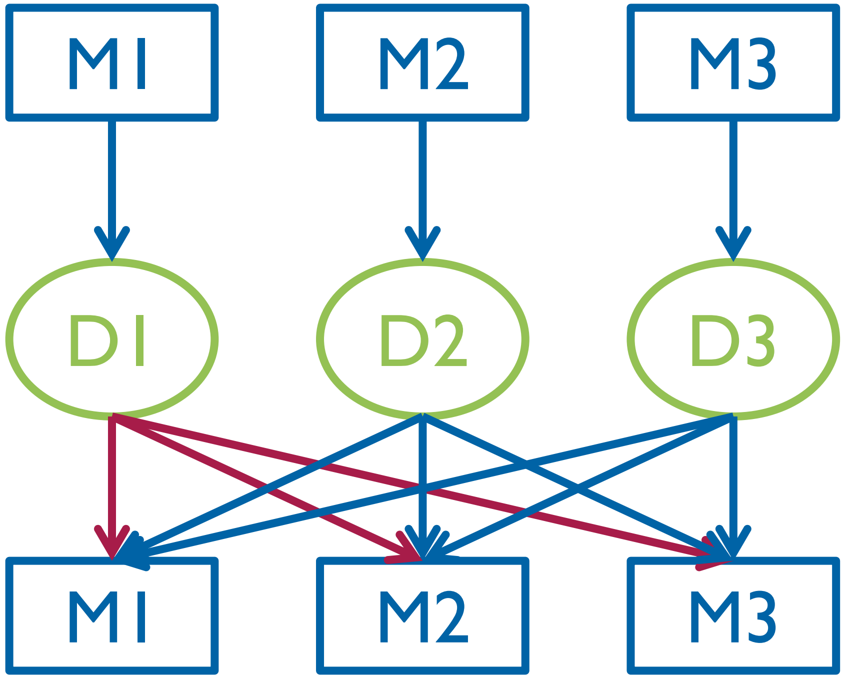


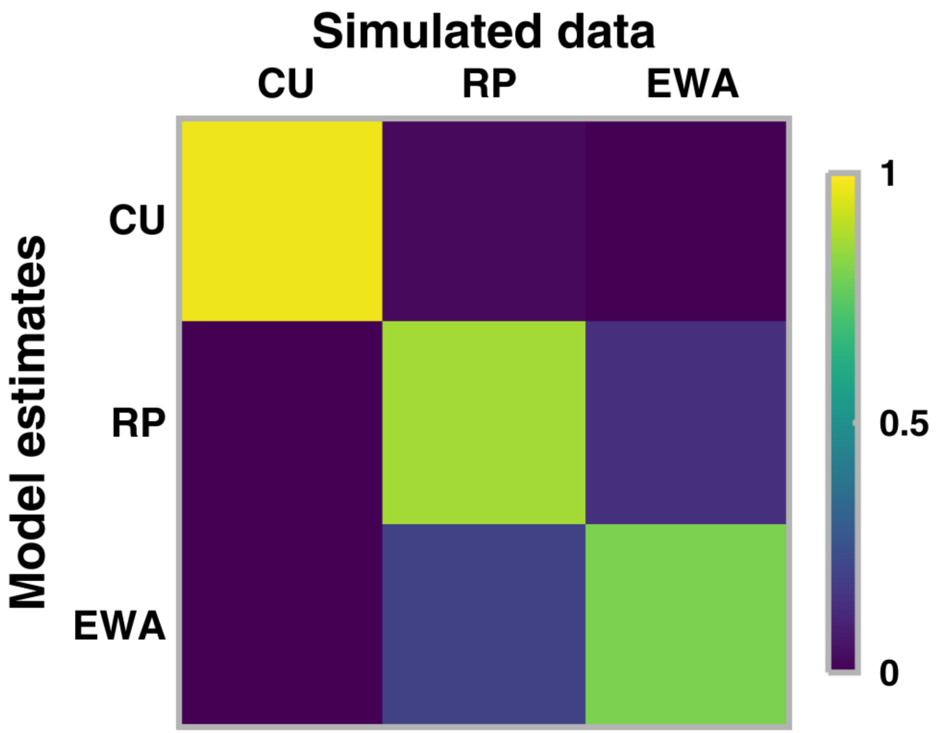
Zhang & Gläscher (2020)

Model recovery:
are models identifiable?

generative process

fitting process





Crawley* & Zhang* et al., (2020)

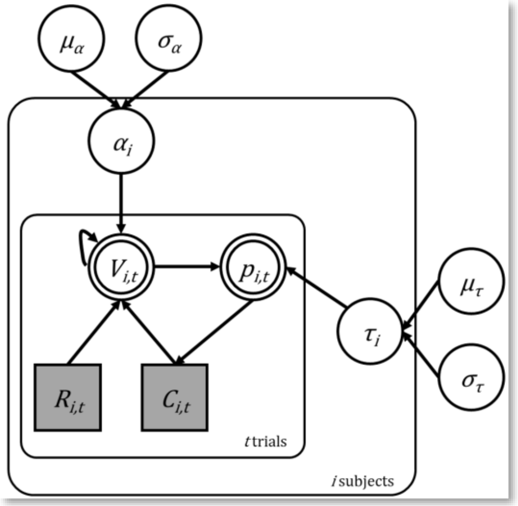
hBayesDM package

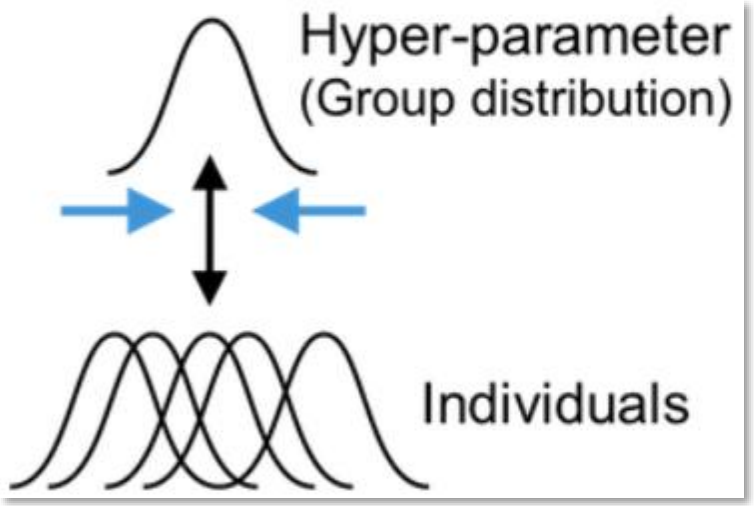
hBayesDM

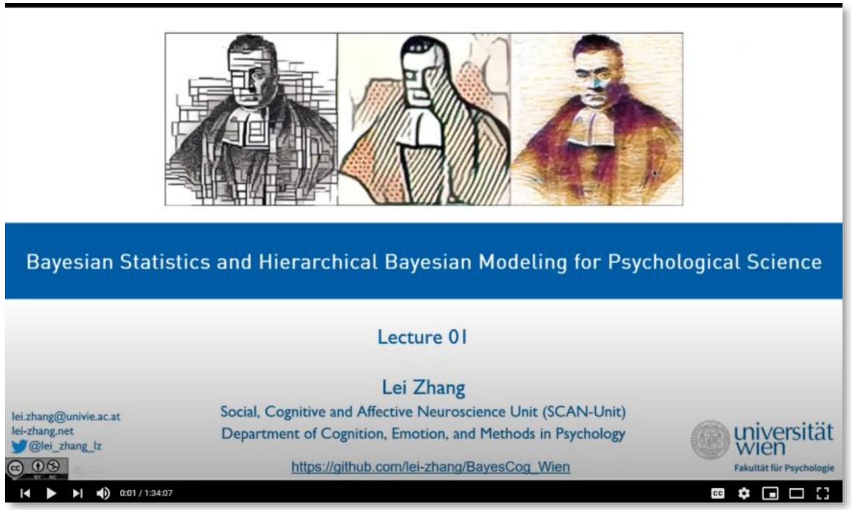
repo status Active | build passing | CRAN 1.0.2 – 2019-11-13 | downloads 33K

DOI 10.1162/CPSY_a_00002

hBayesDM (hierarchical Bayesian modeling of Decision-Making tasks) is a user-friendly package that offers hierarchical Bayesian analysis of various computational models on an array of decision-making tasks. hBayesDM uses Stan for Bayesian inference.







Ahn, Haines, Zhang, (2017)

Adaptive Learning Psychology and Neuroscience

Welcome to BayesCog!
Course overview
Workshop 1: R Basics
Workshop 2: Probability and an introduction to Bayes' theorem
Workshop 3: Building simple models conceptually
Workshop 4: Introduction to building models in Stan
Workshop 5: Bernoulli and linear regression models
Workshop 6: Reinforcement learning models
Workshop 7: Hierachical Bayesian modeling
Workshop 8: Model comparison
Workshop 9: Debugging in Stan
Bonus workshop: Introduction to model-based fMRI
Acknowledgements
Resources

Visitors 1 / 119 | contributors 2 | last commit july | repo size 360.2 MiB

Welcome

This website is an adapted version of teaching materials originally made for the award winning* **BayesCog** seminar at the Faculty of Psychology, University of Vienna, as part of the Advanced Seminar for master's students (Mind and Brain track; recorded during Summer Term 2020/2021). Further content from the BayesCog workshop at UKE Hamburg (2023) have also been added[1].

Recording: Recordings from the original version of the course are available on YouTube (also see below). The most recent recording from the 2021 summer semester is also available on YouTube.

Outreach: **Twitter thread** (being **liked 700+ times** on Twitter) summarizing the contents of the course.

Award/Recognition: The original course received a commendation award from the Society for the Improvement of Psychological Science (SIPS) (also see a tweet), as well as an ECR Teaching Award from the Faculty of Psychology, University of Vienna.

JOSE
The Journal of Open Source Education

BayesCog: A freely available course in Bayesian statistics and hierarchical Bayesian modeling for psychological science

Lei Zhang [1,2,3] **and Aamir Sohail** [1,2]

1 Centre for Human Brain Health, School of Psychology, University of Birmingham, Birmingham, UK **2** Institute for Mental Health, School of Psychology, University of Birmingham, Birmingham, UK **3** Centre for Developmental Science, School of Psychology, University of Birmingham, Birmingham, UK

https://alpn-lab.github.io/BayesCog/

Recommended reading: tutorial

Using reinforcement learning models in social neuroscience: frameworks, pitfalls and suggestions of best practices

Lei Zhang, Lukas Lengersdorff, Nace Mikus, Jan Gläscher, Claus Lamm Author Notes

Social Cognitive and Affective Neuroscience, Volume 15, Issue 6, June 2020, Pages 695–707, https://doi.org/10.1093/scan/nsaa089

https://doi.org/10.1093/scan/nsaa089

Ten simple rules for the computational modeling of behavioral data

Robert C Wilson, Anne GE Collins
University of Arizona, United States; University of California, Berkeley, United States

https://elifesciences.org/articles/49547

ACCEPTED MANUSCRIPT

Computational modelling of social cognition and behaviour—a reinforcement learning primer

Patricia L Lockwood, Miriam Klein-Flügge

Social Cognitive and Affective Neuroscience, nsaa040, https://doi.org/10.1093/scan/nsaa040
Published: 30 March 2020 **Article history** ▾

https://doi.org/10.1093/scan/nsaa040

The Importance of Falsification in Computational Cognitive Modeling

Stefano Palminteri,[1,2,*,‡] Valentin Wyart,[1,2,*,‡] and Etienne Koechlin[1,2,*]

https://doi.org/10.1016/j.tics.2017.03.011

Recommended reading: empirical work


Catecholaminergic challenge uncovers distinct Pavlovian and instrumental mechanisms of motivated (in)action

Jennifer C Swart, Monja I Froböse, Jennifer L Cook, Dirk EM Geurts, Michael J Frank, Roshan Cools, Hanneke EM den Ouden
Radboud University, The Netherlands; University of Birmingham, United Kingdom; Radboud University Medical Center, The Netherlands; Linguistic and Psychological Sciences, Brown University, United States; Brown University, United States

https://elifesciences.org/articles/22169


COGNITIVE NEUROSCIENCE

A brain network supporting social influences in human decision-making

Lei Zhang[1,2*] and Jan Gläscher[1*†]

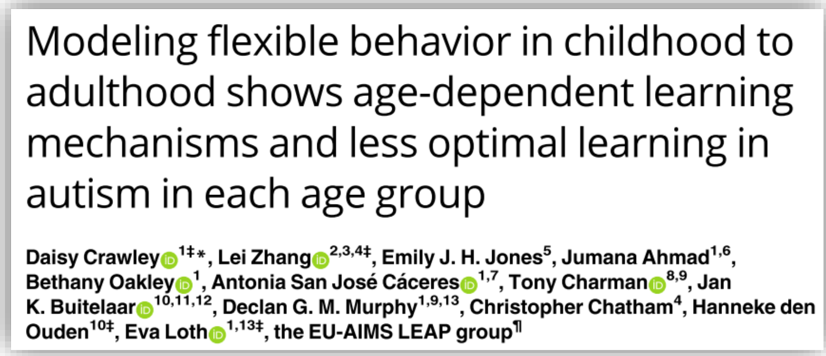
https://advances.sciencemag.org/content/6/34/eabb4159


Social threat learning transfers to decision making in humans

Björn Lindström[a,b,c,1], Armita Golkar[c,d], Simon Jangard[c], Philippe N. Tobler[b], and Andreas Olsson[c]

[a]Department of Social Psychology, University of Amsterdam, 1018 WT Amsterdam, The Netherlands; [b]Laboratory for Social and Neural Systems Research, Department of Economics, University of Zürich, 8001 Zürich, Switzerland; [c]Section for Psychology, Department of Clinical Neuroscience, Karolinska Institutet, 171 77 Stockholm, Sweden; and [d]Department of Clinical Psychology, University of Amsterdam, 1018 WT Amsterdam, The Netherlands

https://www.pnas.org/content/116/10/4732.abstract


Modeling flexible behavior in childhood to adulthood shows age-dependent learning mechanisms and less optimal learning in autism in each age group

Daisy Crawley[1‡*], Lei Zhang[2,3,4‡], Emily J. H. Jones[5], Jumana Ahmad[1,6], Bethany Oakley[1], Antonia San José Cáceres[1,7], Tony Charman[8,9], Jan K. Buitelaar[10,11,12], Declan G. M. Murphy[1,9,13], Christopher Chatham[4], Hanneke den Ouden[10‡], Eva Loth[1,13‡], the EU-AIMS LEAP group[¶]

https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000908

Recommended reading: new direction

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

Article | Open access | Published: 02 July 2025

Discovering cognitive strategies with tiny recurrent neural networks

Li Ji-An, Marcus K. Benna & Marcelo G. Mattar

Nature (2025) | Cite this article

34k Accesses | **354** Altmetric | Metrics

https://www.nature.com/articles/s41586-025-09142-4

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

Article | Open access | Published: 02 July 2025

A foundation model to predict and capture human cognition

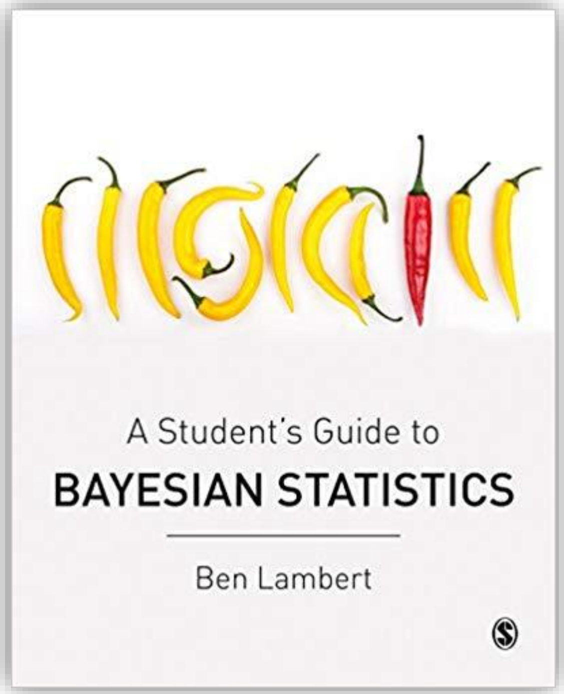
Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K. Eckstein, Noémi Éltető, Thomas L. Griffiths, Susanne Haridi, Akshay K. Jagadish, Li Ji-An, Alexander Kipnis, Sreejan Kumar, Tobias Ludwig, Marvin Mathony, Marcelo Mattar, Alireza Modirshanechi, Surabhi S. Nath, Joshua C. Peterson, Milena Rmus, Evan M. Russek, ... Eric Schulz + Show authors

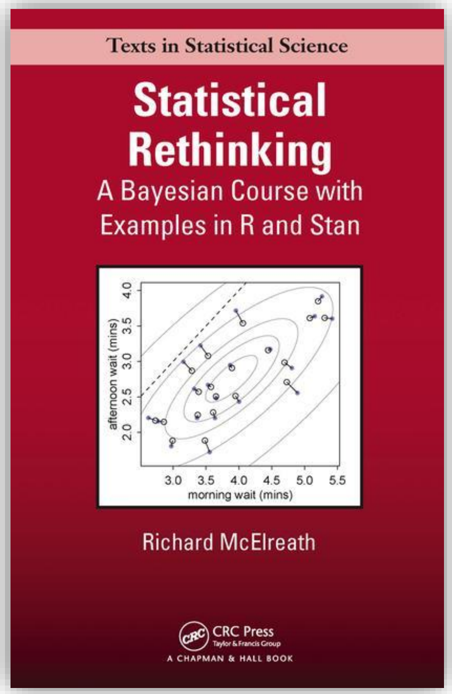
Nature (2025) | Cite this article

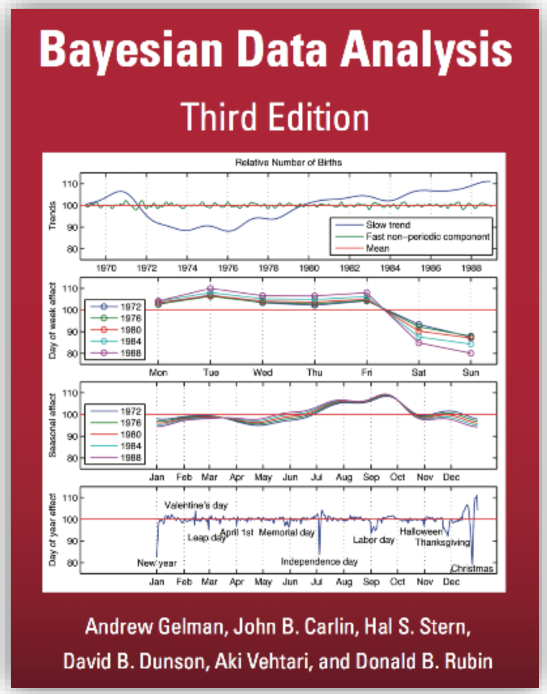
148k Accesses | **1** Citations | **633** Altmetric | Metrics

https://www.nature.com/articles/s41586-025-09215-4

Recommended reading: book







Want to learn more about Stan?

Workshops

- StanCon 2019 Hierarchical Models
- PyData NYC 2019
- StanCon 2018 Intro Stan
- StanCon 2018 Hierarchical Models

https://mc-stan.org/workshops/

Twitter

Richard McElreath
@rlmcelreath

Quiche Lorraine,
@dan_p_simpson

\mathfrak{Michael
@betanalpha

EJ Wagenmakers
@EJWagenmakers

Stan forum

StanCon 2020. August 11-14 at Oregon State University
Announcements stancon
StanCon 2020 will be at Oregon State University! There will be two days of tutorials followed by two days of talks, open discussions, and statistical modeling. Up-to-date details at https://mc-stan.org/events/stancon202… read more

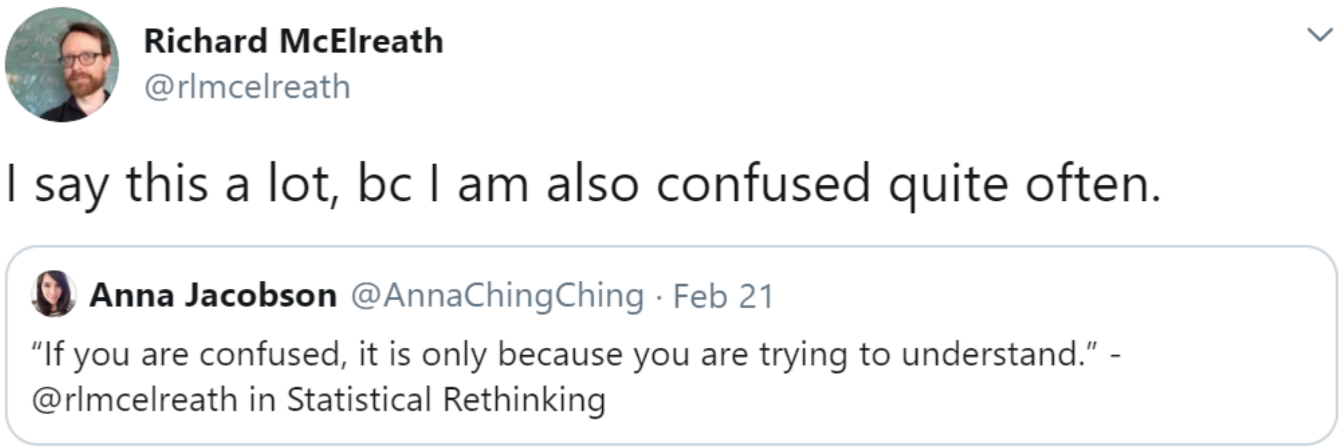
Welcome to the Stan Forums!
The Stan Forums provide a community for asking and answering questions about all aspects of Stan. Before creating a new topic please search the Forums to see if your questions has already been answered, or check out the… read more

Unable to retrieve parameters from a dynamic model
Modeling fitting-issues specification

https://discourse.mc-stan.org/

Summary

- Computational modeling is never new → don't let it fear you!
- Learn some statistics (e.g., different statistical distributions)
- Learn some math (e.g., linear algebra)
- Learn some programing (e.g., R/Python/Julia/Matlab)
- Learn to seek external help (e.g., existing packages)
- Learn in pairs; practice makes perfect!



Acknowledgement

l.zhang.13@bham.ac.uk

https://lei-zhang.net/

@lei_zhang_lz

@LeiZhang

@lei-zhang

Thank you!


ANY
QUESTIONS
?

Happy Computing!