Bayesian Statistics and Hierarchical Bayesian Modeling for Psychological Science

Lecture 10

Lei Zhang

Social, Cognitive and Affective Neuroscience Unit (SCAN-Unit)

Department of Basic Psychological Research and Research Methods

https://github.com/lei-zhang/BayesCog_Wien

lei.zhang@univie.ac.at
lei-zhang.net
@lei_zhang_lz

universität wien
Fakultät für Psychologie

Evaluate the review

- the ability to briefly and clearly summarize the main findings
- critical thinking
  - regarding the design
  - regarding the analyses (stats)
- appreciate the use of cognitive modeling
  - is the modeling approach appropriate to answer the research question?
  - is the interpretation sound?
  - could there be alternative models?
  - model recovery
  - posterior check etc.

New deadline: 17th Jan. 2020

1^{st}	2^{nd}
70%	40%
30%	60%

Programming Project

Deadline: 23rd Feb. 2020

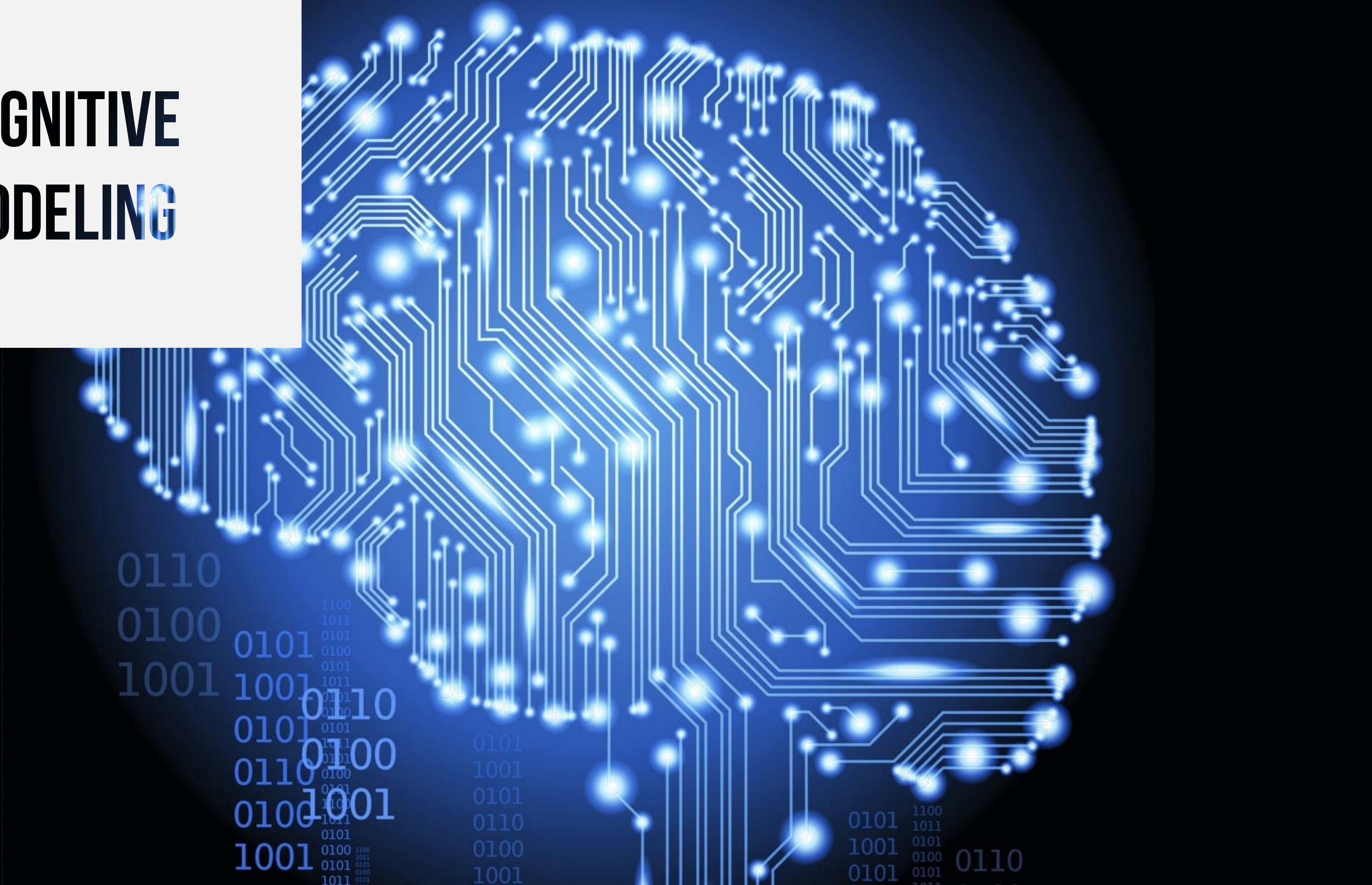
Your colleague Lisa is currently working on a project about intertemporal decision-making. In this experiment, participants were instructed to choose from an immediate but small monetary reward (aka, small but soon option, SS) and a large but delayed alternative (aka, large but late option, LL). See Fig.1 for an example. She just finished the data collection and is now focusing on the analysis. She is aware of a commonly used delay discounting model in such tasks, that is, the Hyperbolic Discounting Model (Mazur, 1987), and would like to fit this model with hierarchical Bayesian methods (Gelman et al., 2013). Her supervisor told her that an increasing number of researchers are using a newly developed programming language Stan (Carpenter et al., 2017) for this purpose and encouraged her to try it out. She knows R already, but she has never used Stan. She spent a few days reading the Stan documentation and coded her first Stan model. As she expected, she received some errors, and she was unable to understand them. She was told that you are now taking a course on Bayesian statistics and cognitive modeling, and therefore, she came to you for some help with the Stan code.

A few days later, her supervisor sent her a paper about a new model for intertemporal choices (Ericson et al., 2015). In this paper, the authors proposed a simple heuristic model, and claimed that this model would outperform traditional delay discounting models. She read the equations a bit, but she had no clue what this paper was talking about. Therefore, she came to you again, asking if you could help her implement this model in Stan.

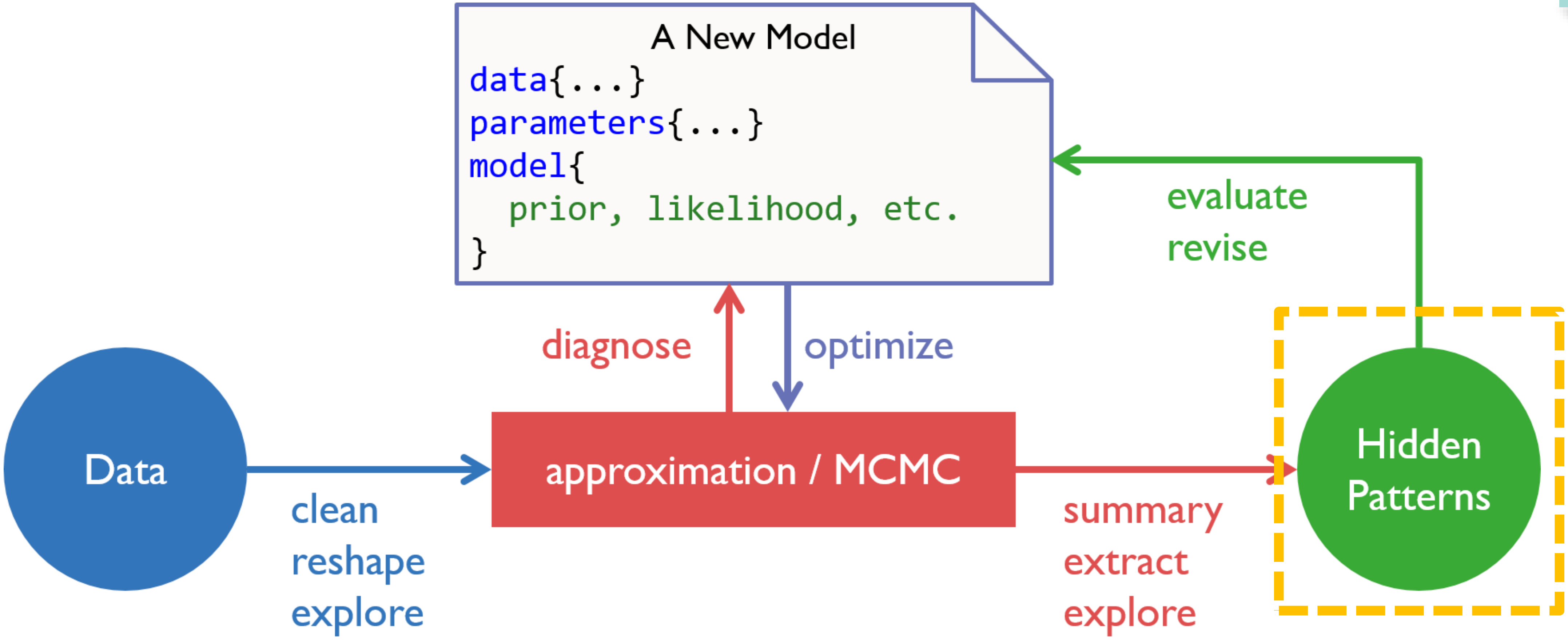
[Task]

1. Fix the potential bugs/errors in the "hyperbolic. stan" model file, and fit this model.
2. Read the (Ericson et al., 2015) paper, implement this model in "heuristic.stan", and fit this model.
3. Compare these two models using widely applicable information criterion (WAIC).

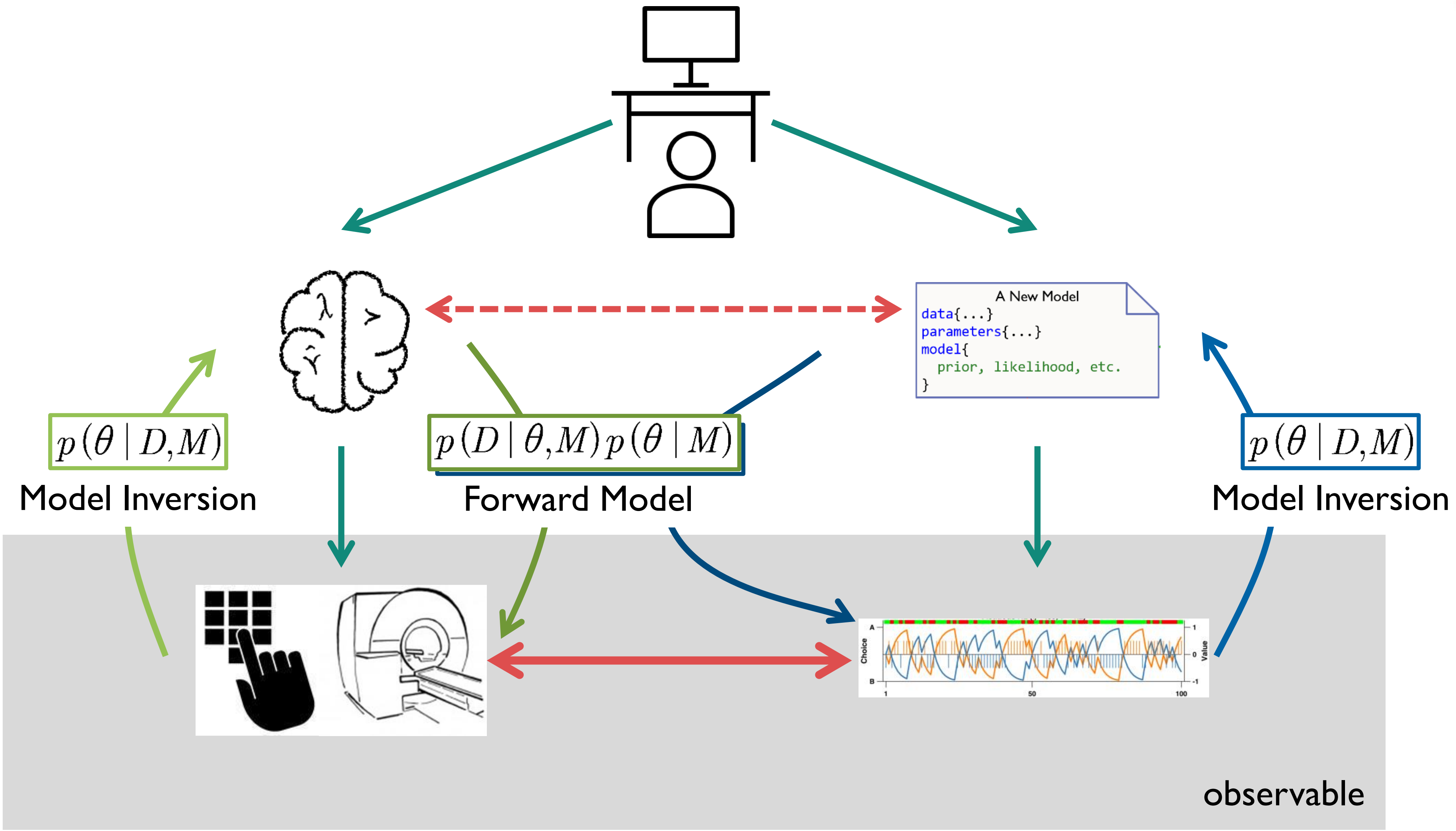
COGNITIVE MODELING



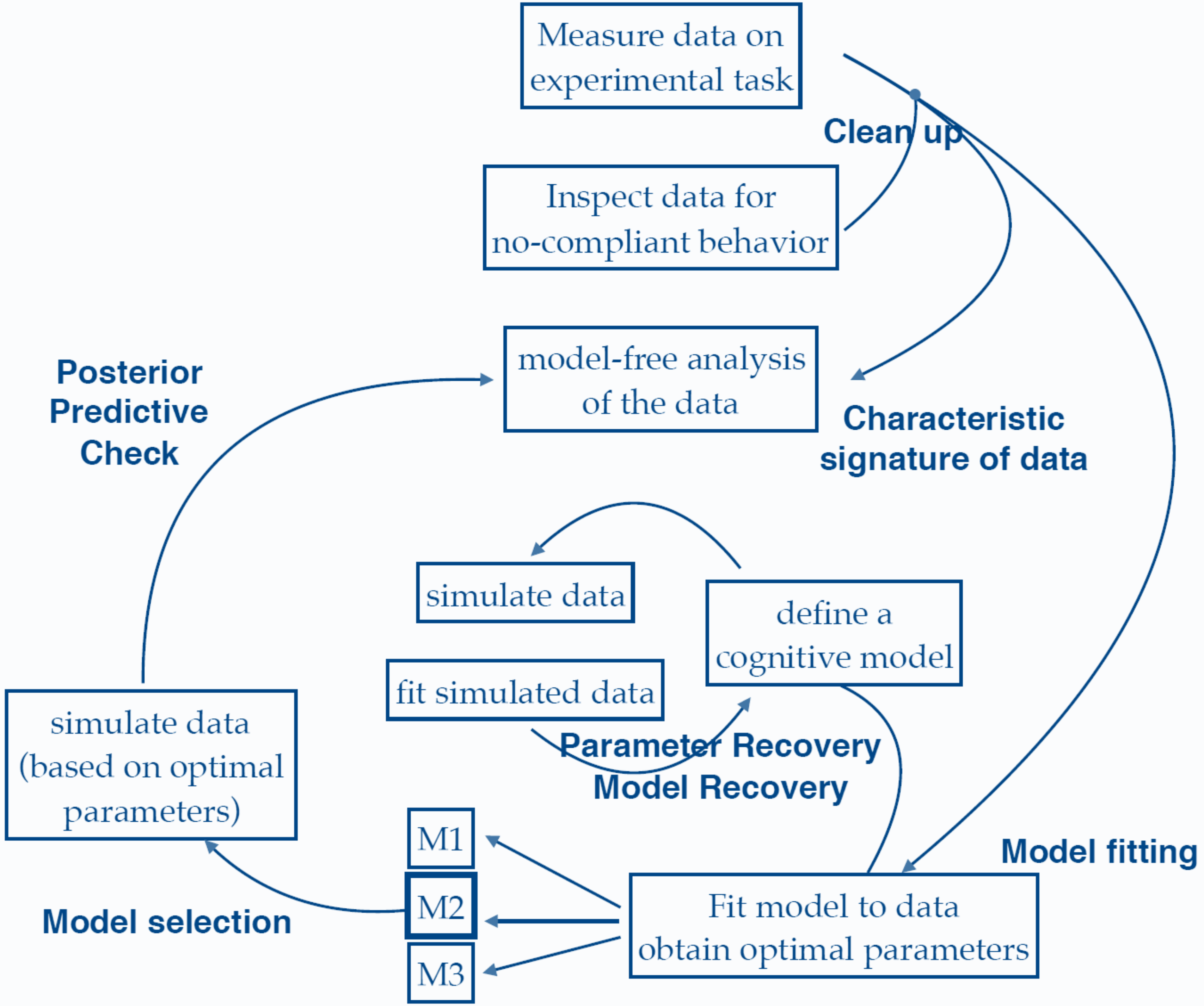
cognitive model
statistics
computing



What is Cognitive Modeling?



Lewandowsky & Farrell (2011); Stephan et al. (2016); Ahn et al., (2017)

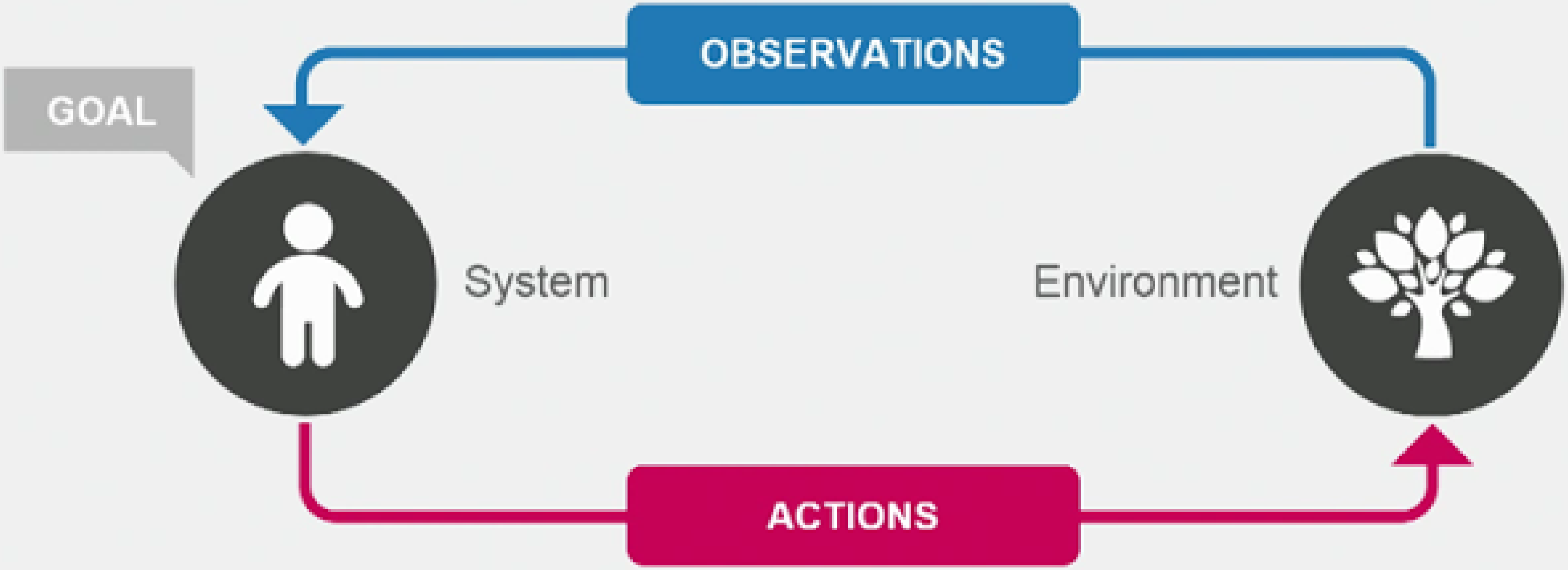

Measure data on experimental task
Clean up
Inspect data for no-compliant behavior
model-free analysis of the data
Characteristic signature of data
Posterior Predictive Check
simulate data
define a cognitive model
fit simulated data
Parameter Recovery
Model Recovery
simulate data (based on optimal parameters)
M1
M2
M3
Model selection
Model fitting
Fit model to data obtain optimal parameters

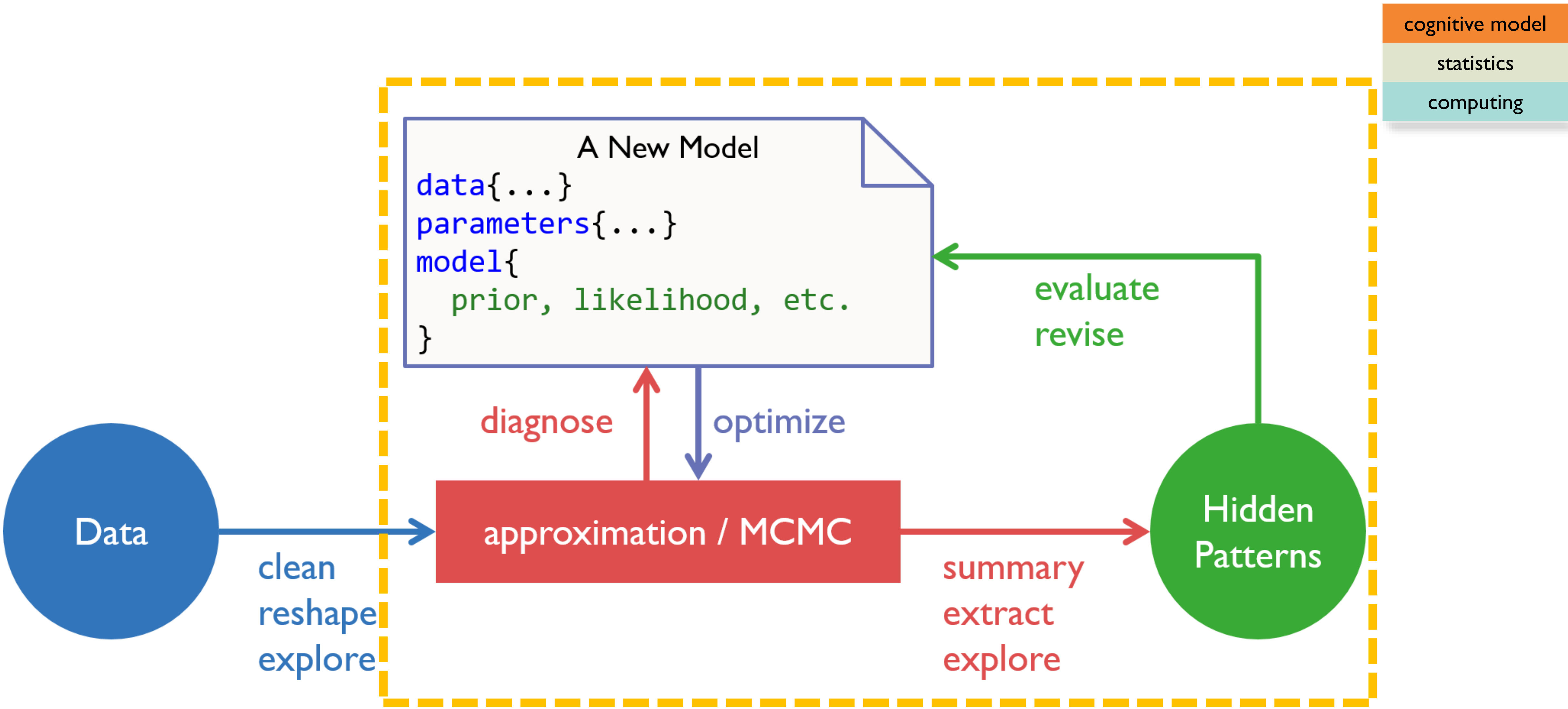
Essentially, all the models are wrong, but some are useful.

– George E. P. Box

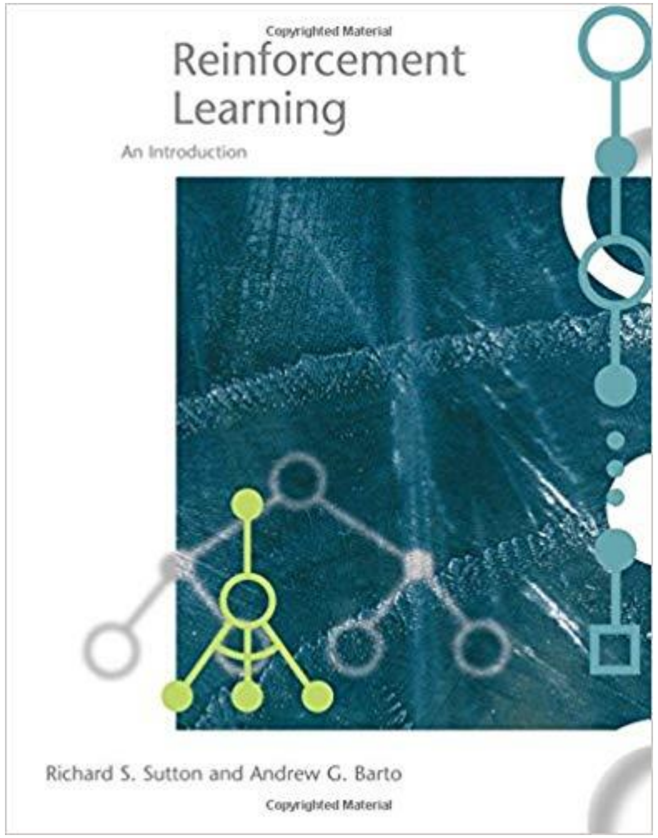
Essentially, all the models are ~~wrong~~ imperfect, but some are useful.

REINFORCEMENT LEARNING FRAMEWORK

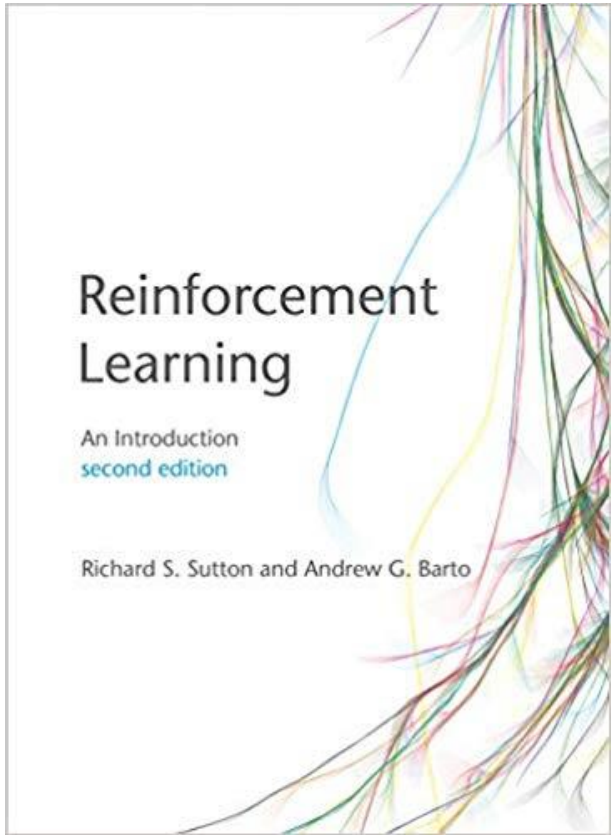



cognitive model
statistics
computing
A New Model
data{...}
parameters{...}
model{
prior, likelihood, etc.
}
evaluate
revise
diagnose
optimize
Data
approximation / MCMC
Hidden Patterns
clean
reshape
explore
summary
extract
explore

The very short history

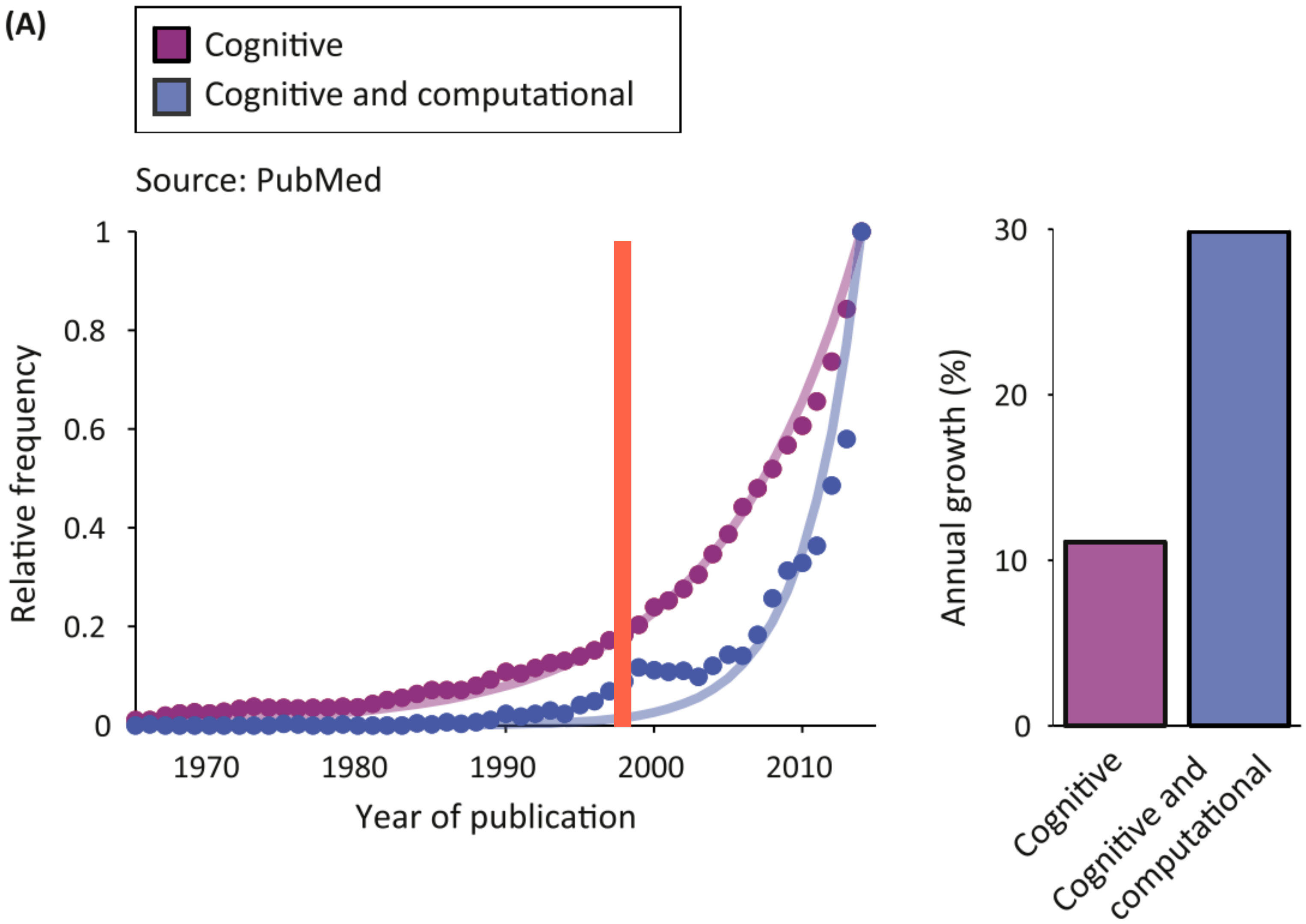


1998



2018

Boom in Cognitive Modeling



Palminteri et al. (2017)

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

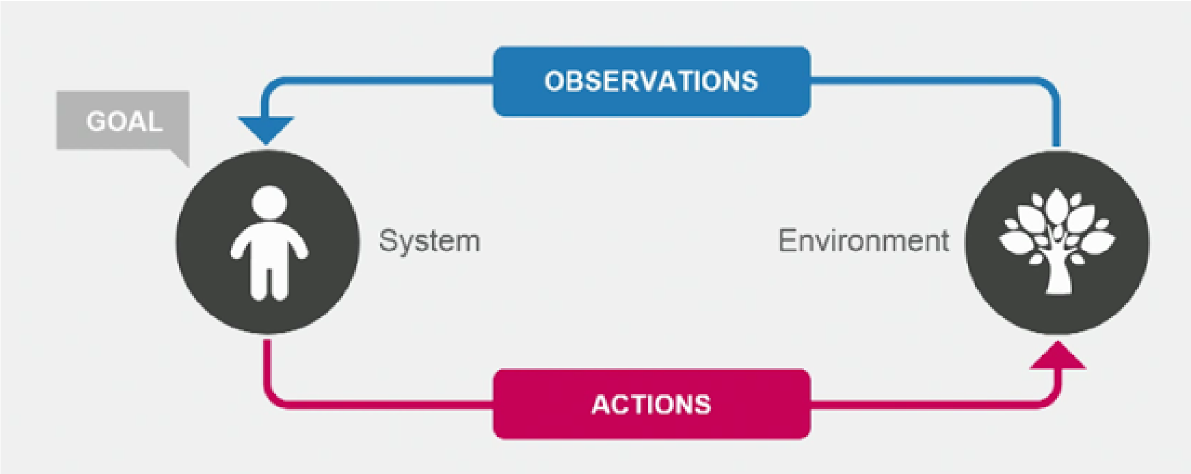
What can be your strategies:

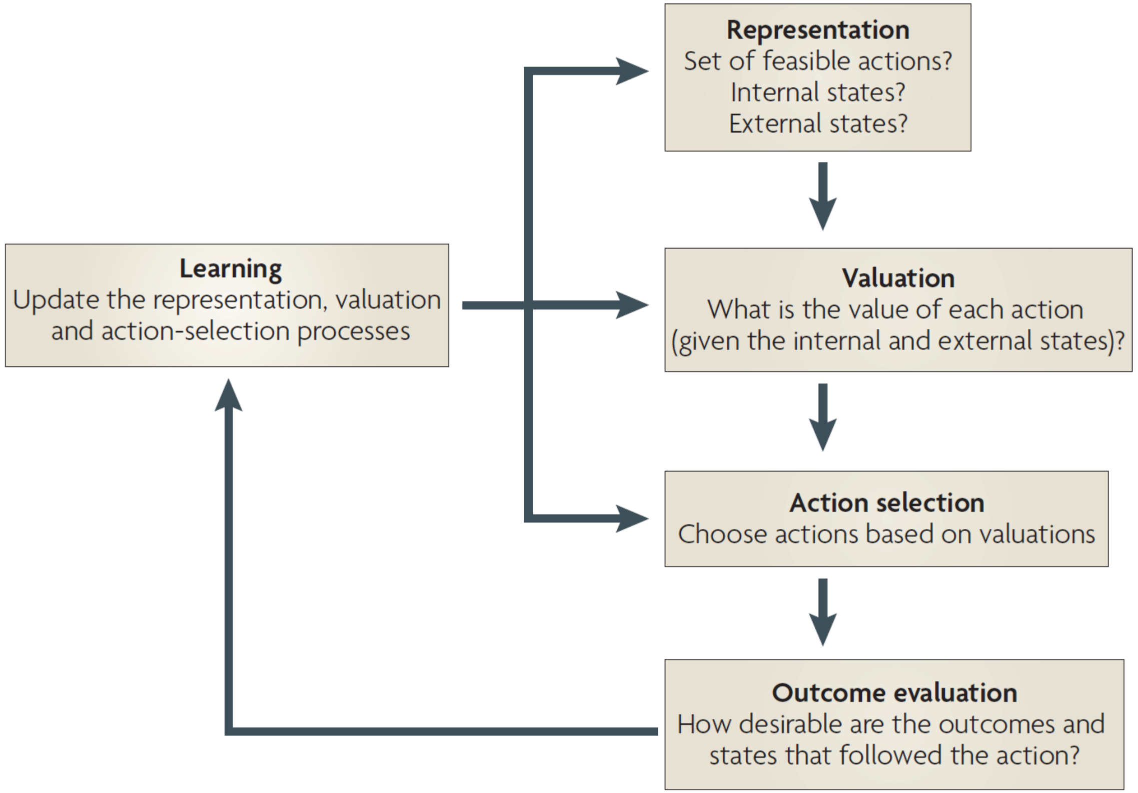
1. predict the value of each deck
2. choose the best
3. learn from outcome to update predictions

(repeat)

How prediction is shaped by learning?

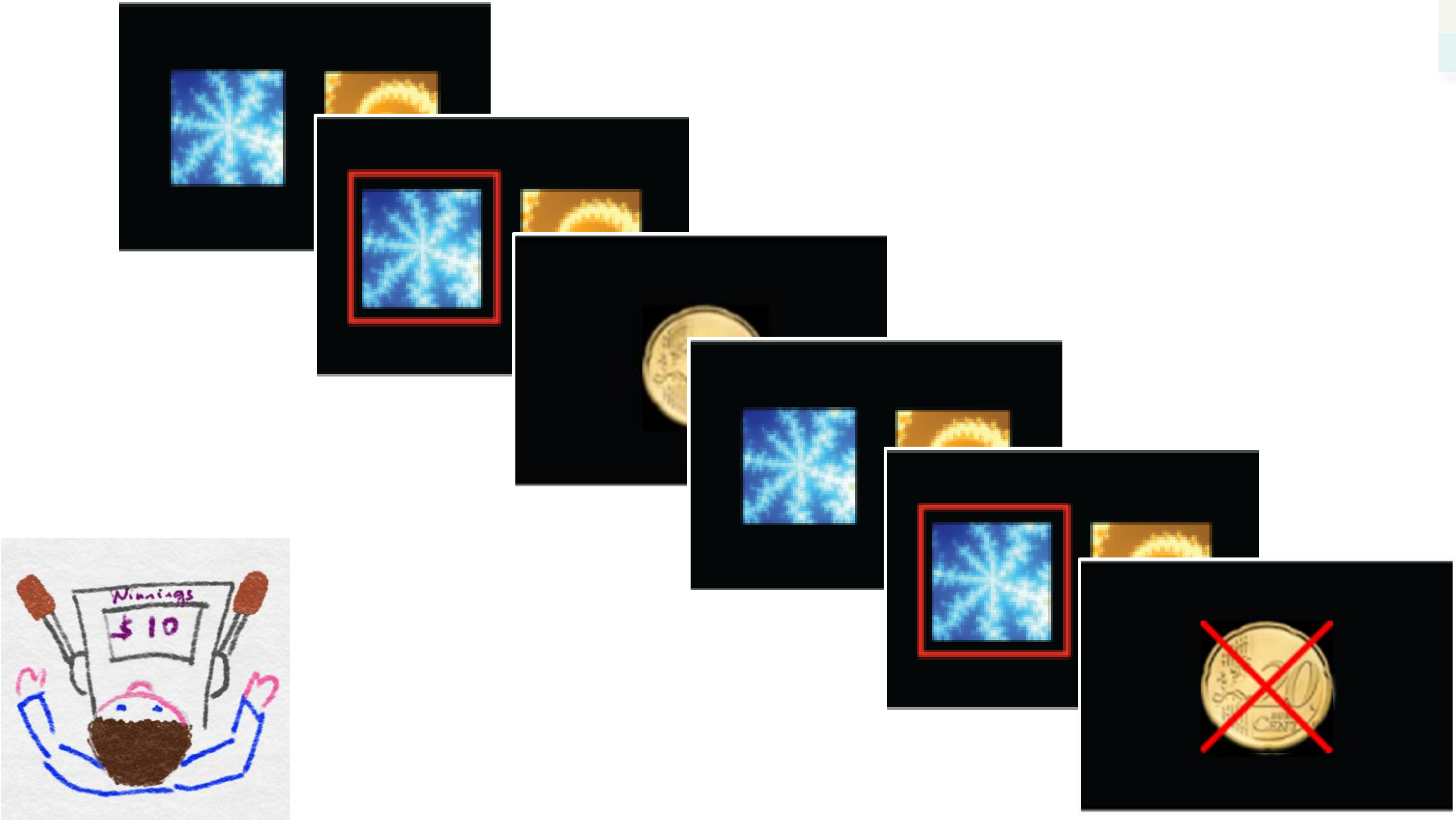
cognitive model
statistics
computing



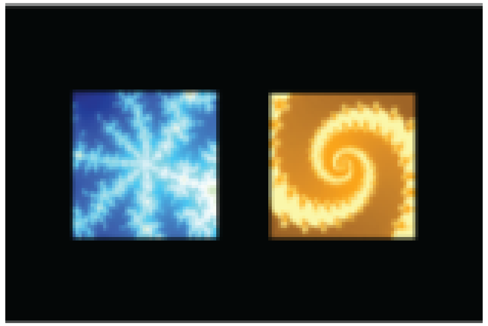


Sutton & Barto (1998); Rangel et al (2008); [Daw & Niv workshop 2015]

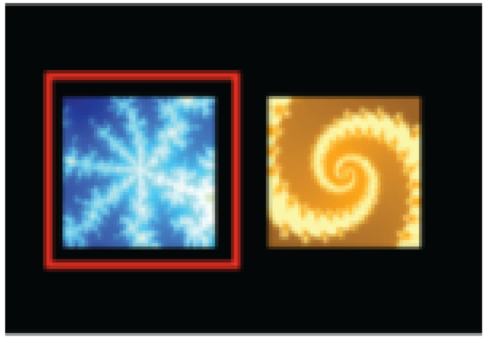
cognitive model
statistics
computing



One simple experiment: two choice task



choice presentation



action selection

outcome

what do we know?

what can we measure?

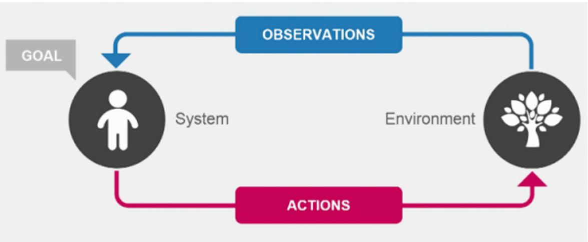
what do we not know?

Data: choice & outcome

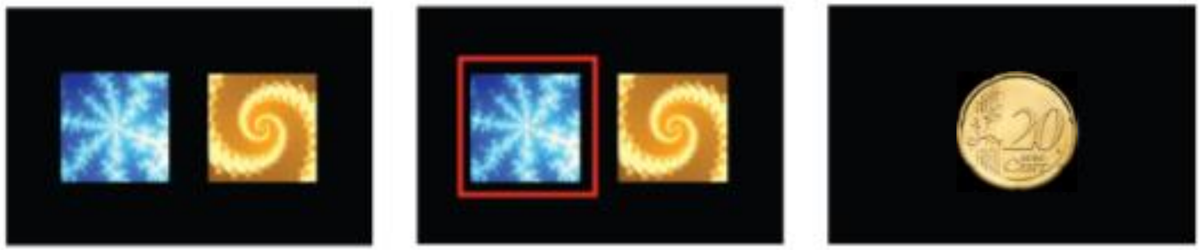
Summary stats: choice accuracy

Learning algorithm: RL update

p(choosing the better option)



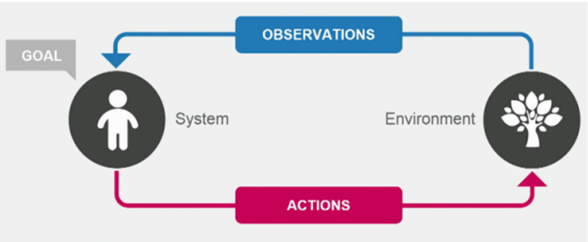
Rescorla-Wagner Value Update



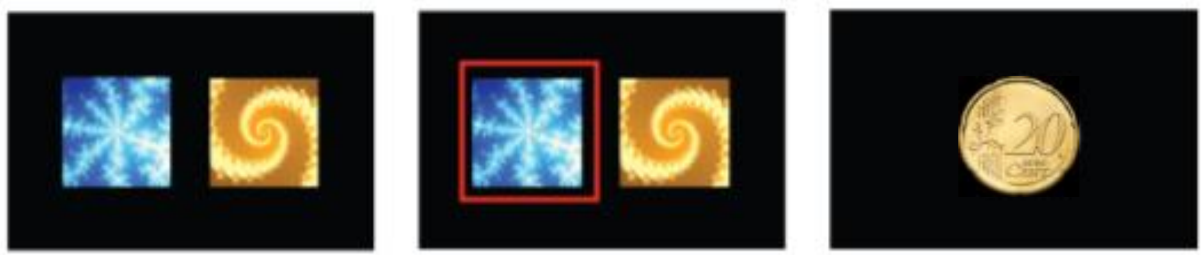
Cognitive Model
- cognitive process
- using internal variables and free parameters

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

Rescorla & Wagner (1972)



Rescorla-Wagner Value Update



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

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

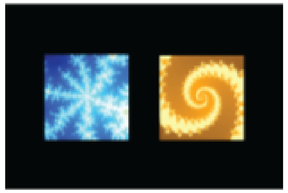
α – learning rate
PE – reward prediction error
V – value
R – reward

Rescorla & Wagner (1972)

Understand the learning rate

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

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



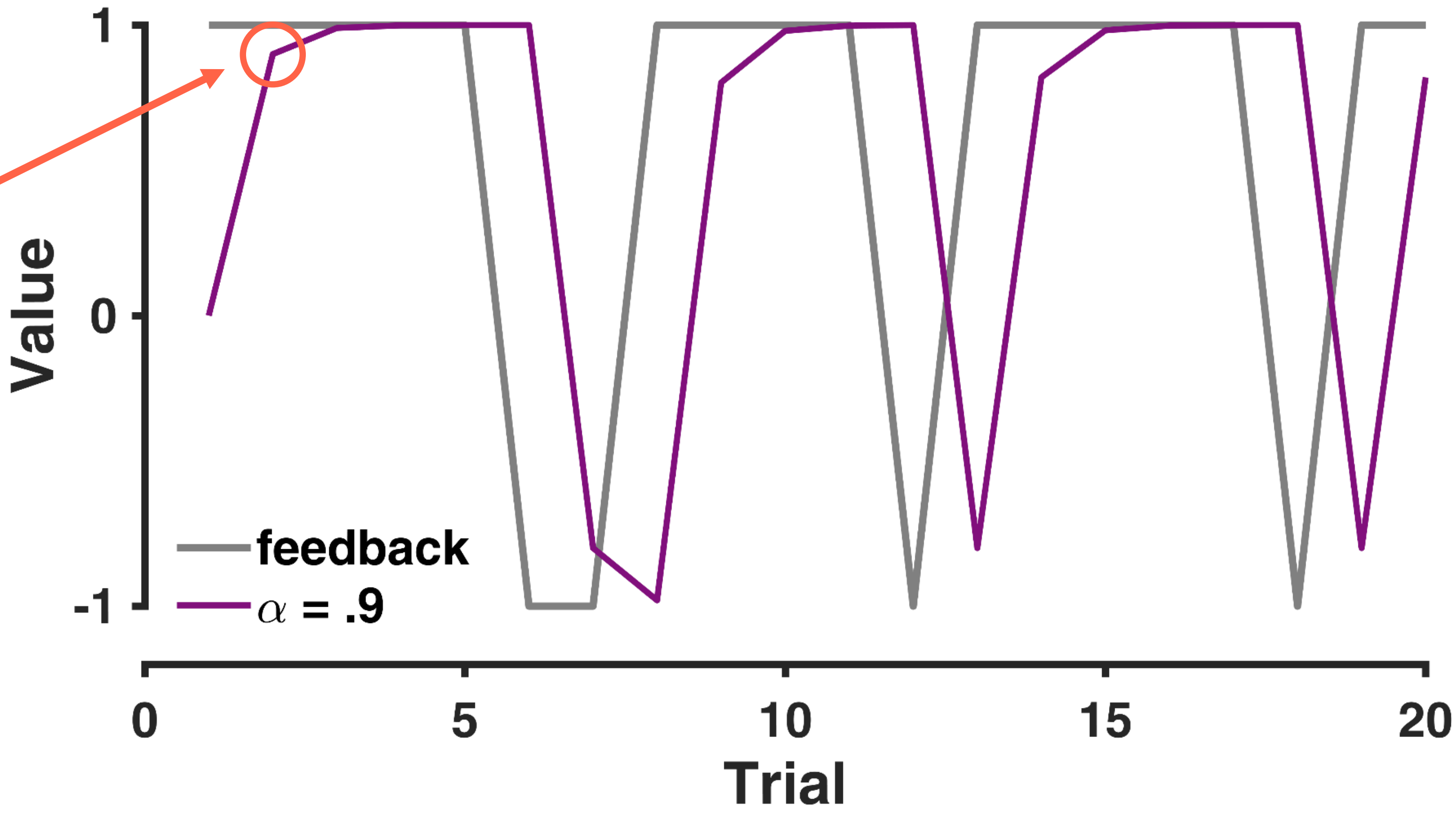
if $\alpha = 0.9$

$V_1 = 0$

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

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

$= 0.9$

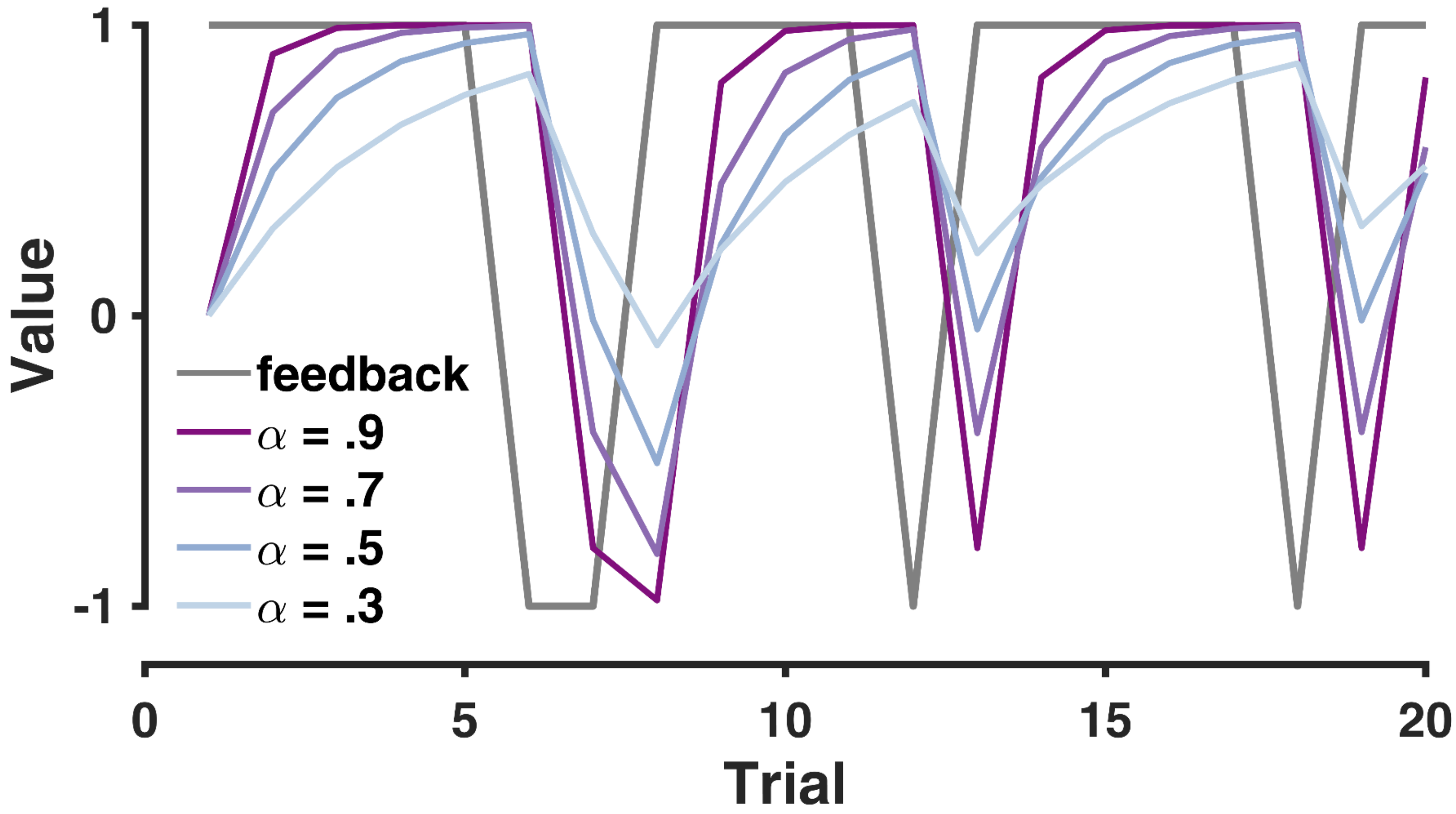


reward contingency – 80:20

Understand the learning rate

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

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



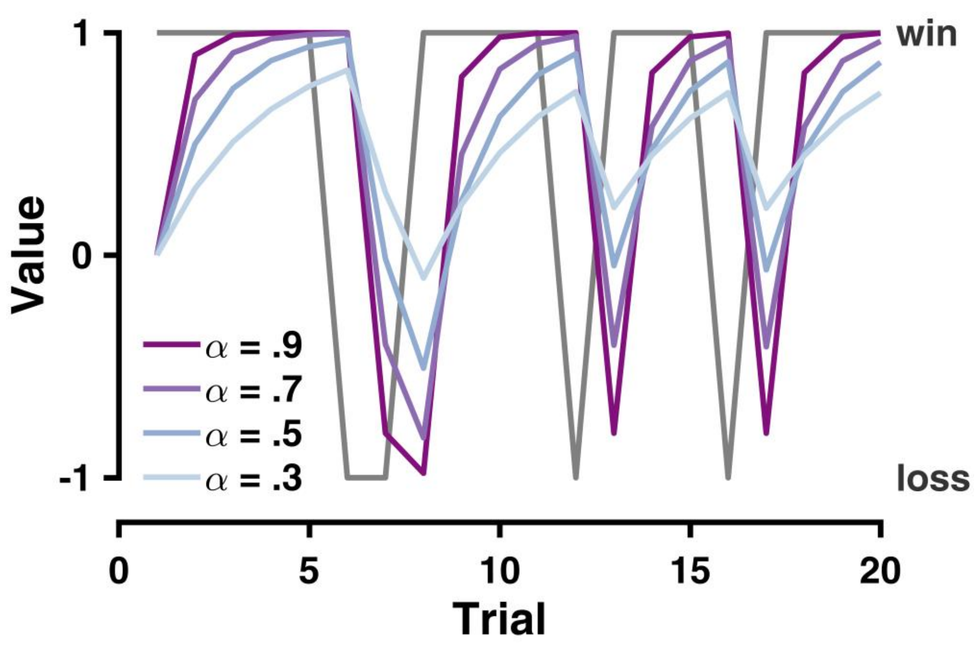
reward contingency – 80:20

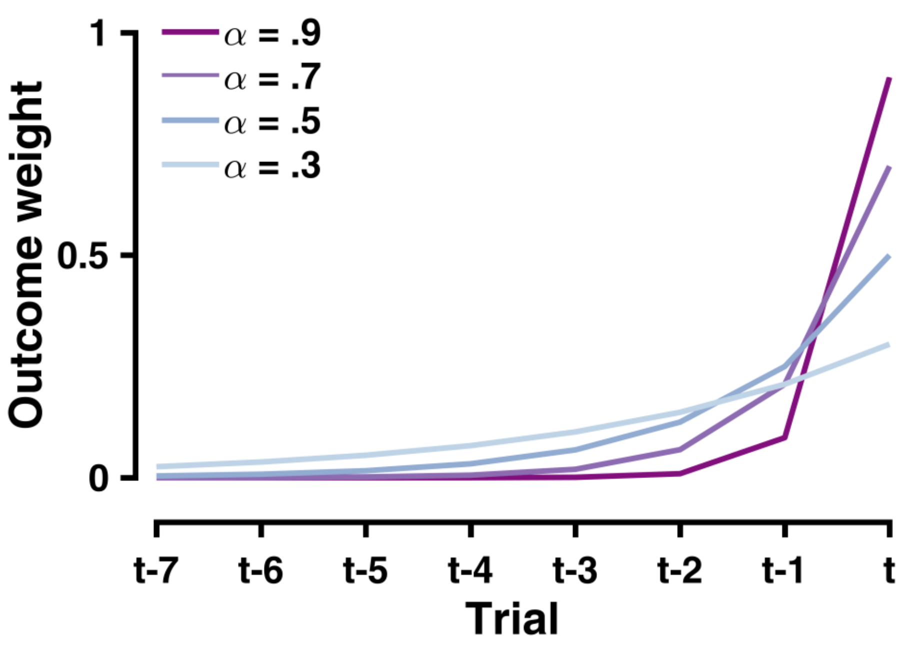
Understand the learning rate

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

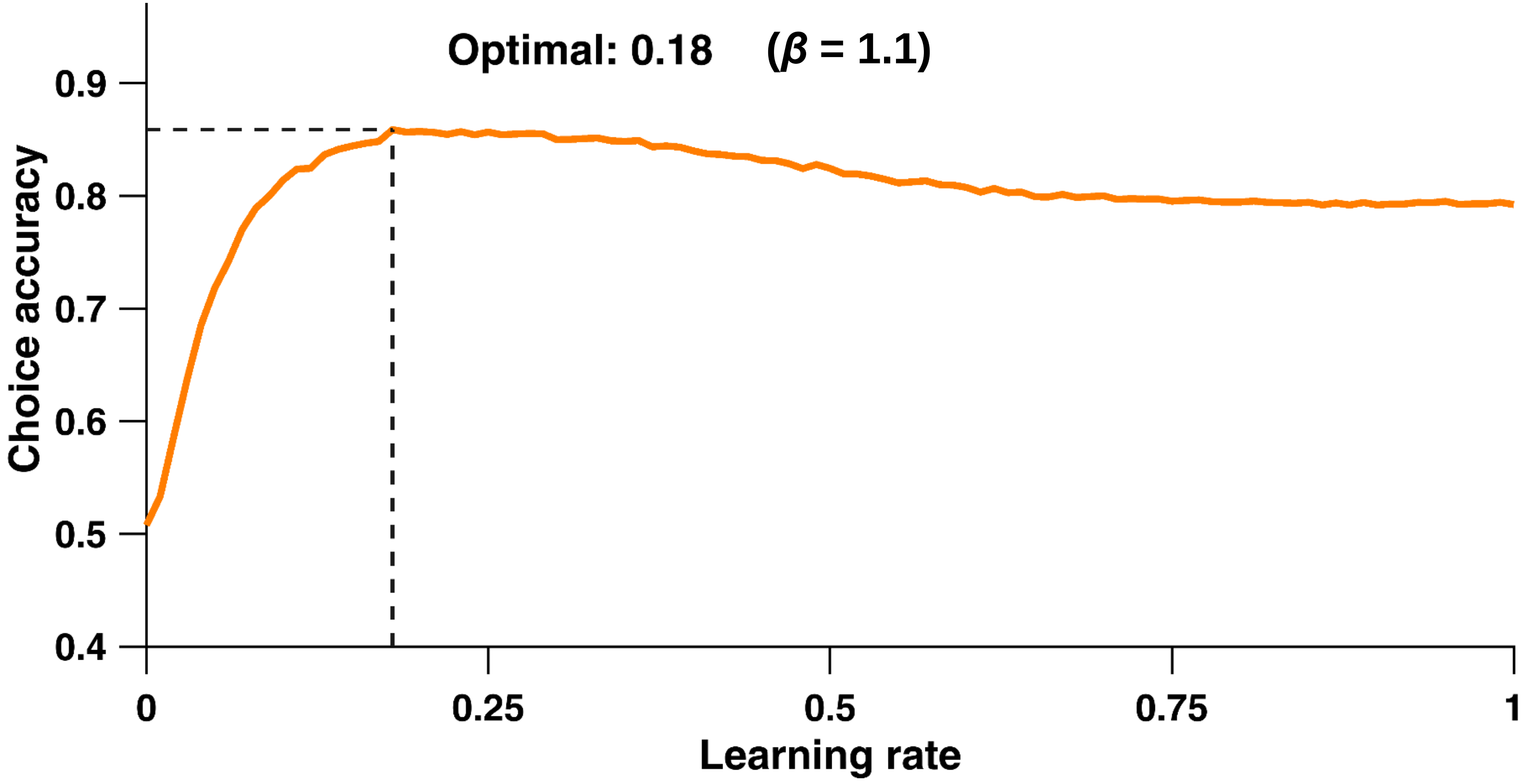
Prediction error: $\mathrm{PE}_{t-1} = R_{t-1} - V_{t-1}$

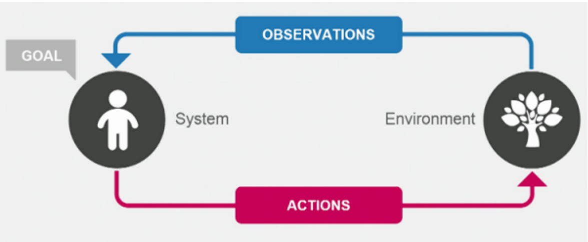
$$\begin{aligned} V_t &= (1-\alpha)V_{t-1} + \alpha R_{t-1} \\ &= (1-\alpha)(V_{t-2} + \alpha(R_{t-2} - V_{t-2})) + \alpha R_{t-1} \\ &= (1-\alpha)^t V_0 + \boxed{\sum_{i=1}^{t}(1-\alpha)^{t-i}\alpha R_i} \end{aligned}$$



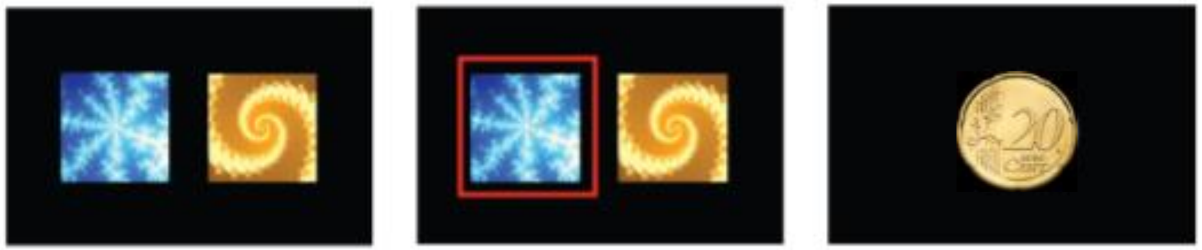


Optimal learning rate?





Rescorla-Wagner Value Update



Value update:

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

Prediction error:

$$PE_t = R_t - V_t$$

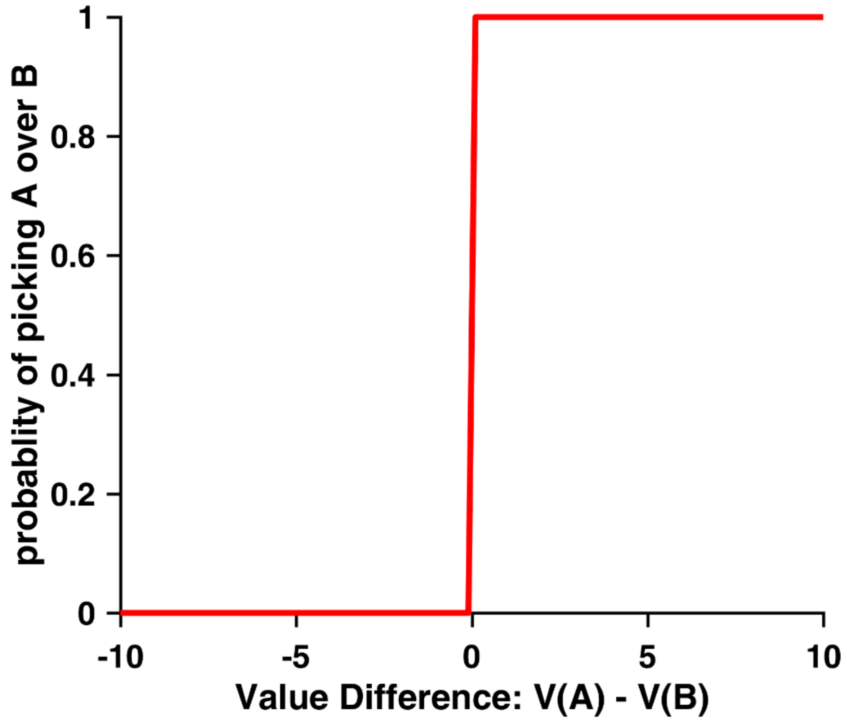
choice rule:

greedy / ε-greedy / softmax

Rescorla & Wagner (1972)

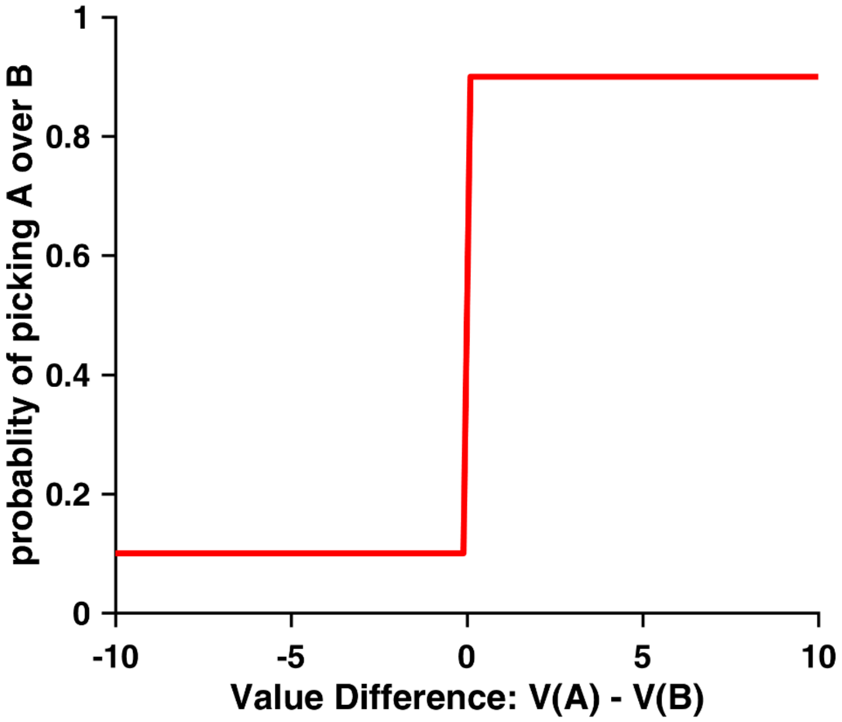
Choice rule: greedy

$$p(C=a)=\begin{cases}1, V(a)>V(b)\\0, V(a)<V(b)\end{cases}$$

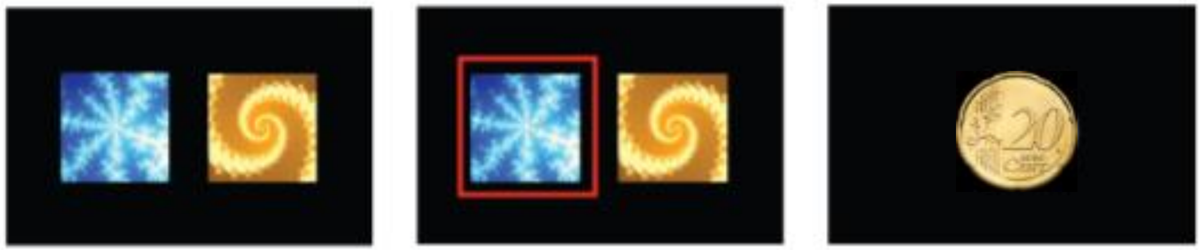


Choice rule: ε-greedy

$$p(C=a) = \begin{cases} 1-\varepsilon, & V(a) > V(b) \\ \varepsilon, & V(a) < V(b) \end{cases}$$



Choice rule: softmax

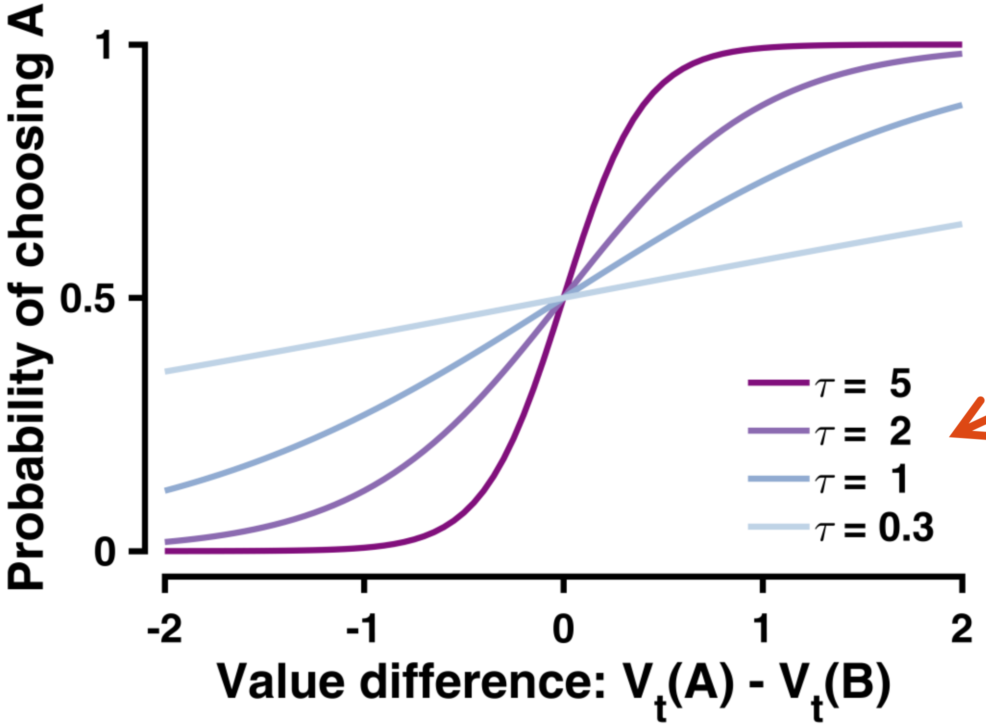


V()

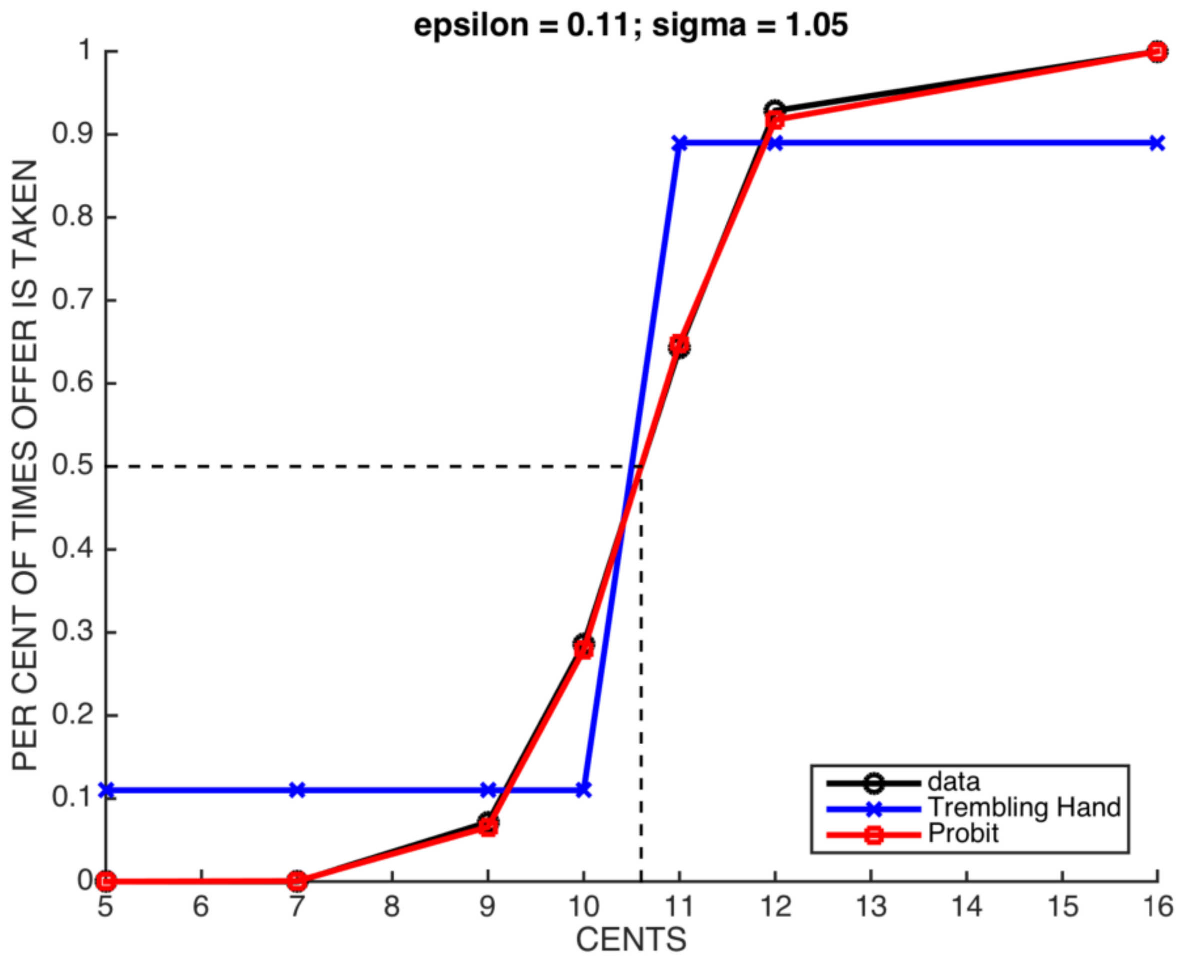
V()

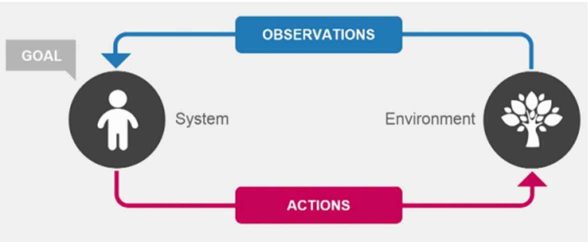
$$\boxed{p_t(A) = \frac{e^{\tau * V_t(A)}}{e^{\tau * V_t(A)} + e^{\tau * V_t(B)}}}$$

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

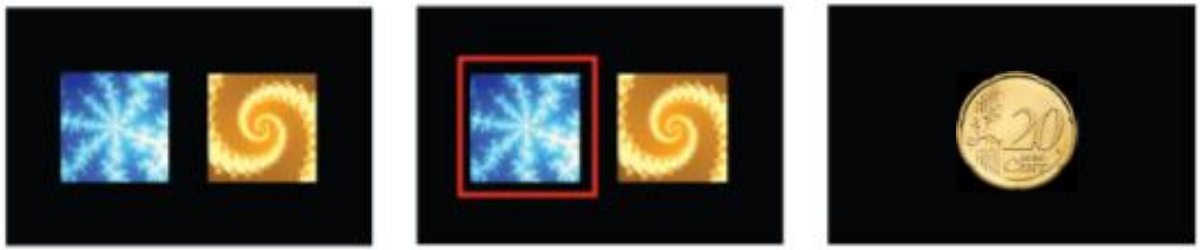


Choice rule: direct comparison





Rescorla-Wagner Value Update



Value update:

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

Prediction error:

$$PE_t = R_t - V_t$$

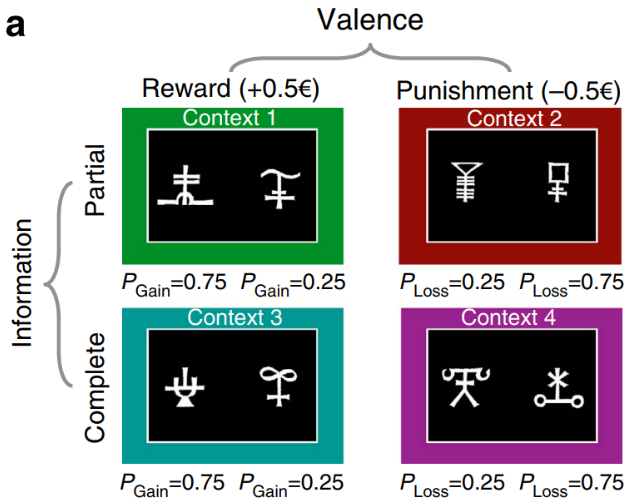
choice rule (sigmoid /softmax):

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

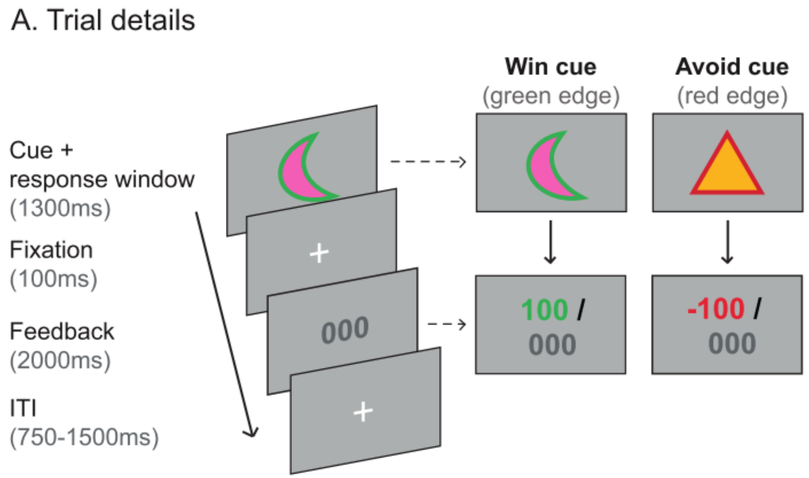
α - learning rate
PE - reward prediction error
V - value
R - reward
τ - softmax temperature

Rescorla & Wagner (1972)

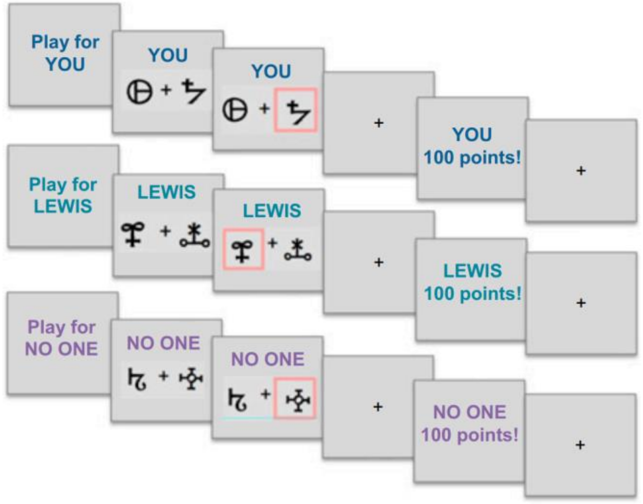
Generalizing RL framework



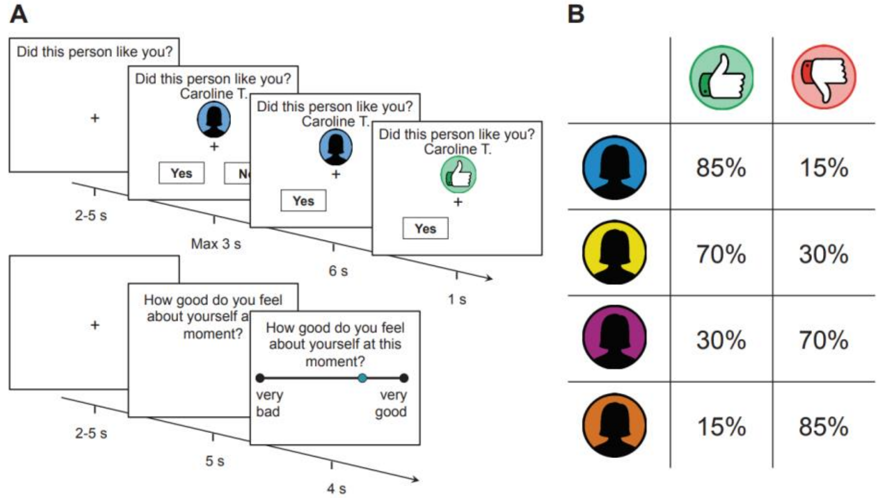
Palminteri et al. (2015)



Swart et al. (2017)



Lockwood et al. (2016)



Will et al. (2017)

Bayesian Statistics and Hierarchical Bayesian Modeling for Psychological Science

Lecture 11

Lei Zhang

Social, Cognitive and Affective Neuroscience Unit (SCAN-Unit)

Department of Basic Psychological Research and Research Methods

lei.zhang@univie.ac.at

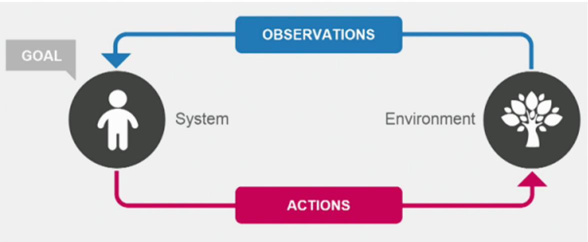
lei-zhang.net

@lei_zhang_lz

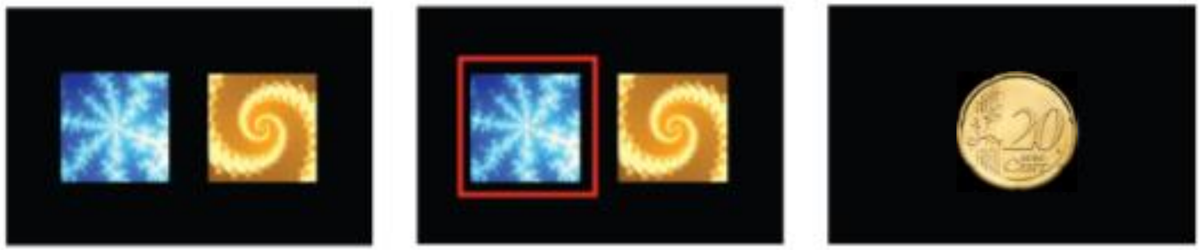
universität wien

Fakultät für Psychologie

https://github.com/lei-zhang/BayesCog_Wien



Rescorla-Wagner Value Update



Value update:

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

Prediction error:

$$PE_t = R_t - V_t$$

choice rule (sigmoid /softmax):

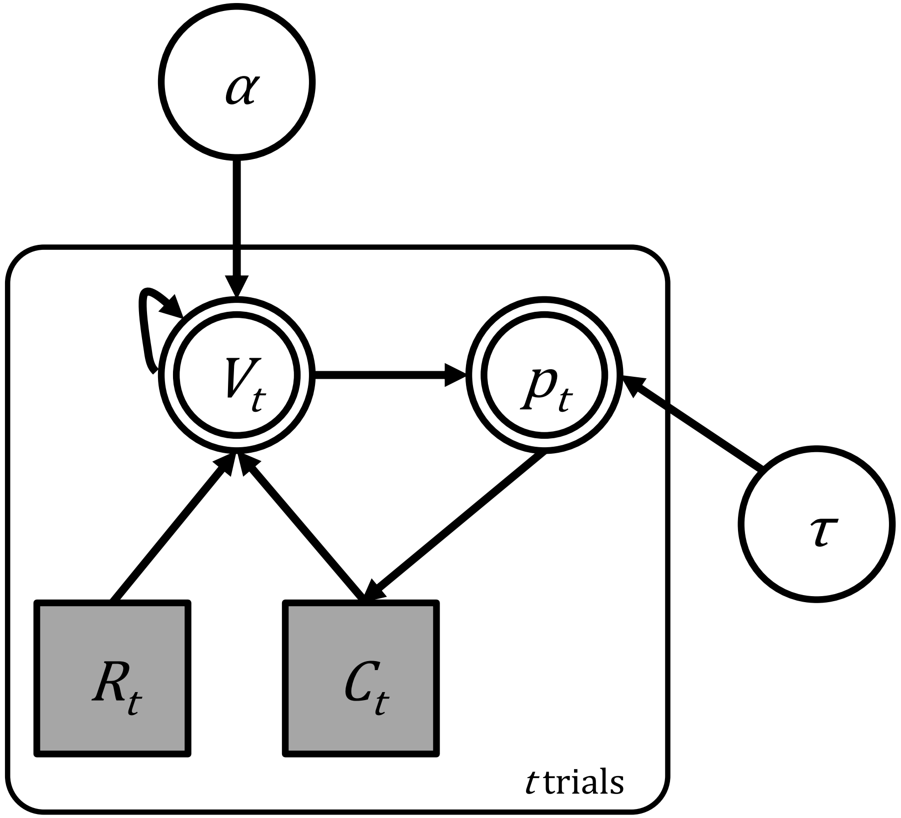
$$p(C{=}a) = \frac{1}{1+e^{\tau*(v(b)-v(a))}}$$

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

Rescorla & Wagner (1972)

RL – Implementation

cognitive model
statistics
computing



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

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

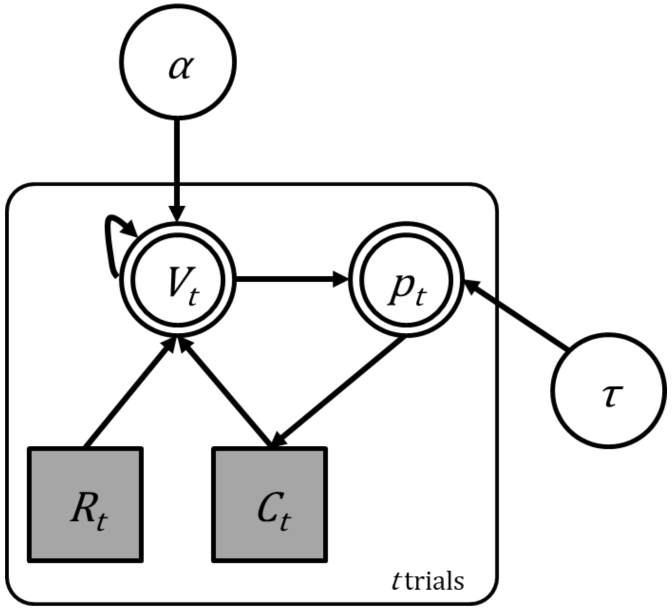
$$p_t(C=A) = \frac{1}{1+e^{\tau(V_t(B)-V_t(A))}}$$

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

deterministic variable

RL – Implementation

cognitive model
statistics
computing



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

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

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

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

```
transformed data {
  vector[2] initV;
  initV = rep_vector(0.0, 2);
}

model {
  vector[2] v[nTrials+1];
  real pe[nTrials];

  v[1] = initV;

  for (t in 1:nTrials) {
    choice[t] ~ categorical_logit( tau * v[t] );

    pe[t] = reward[t] - v[t,choice[t]];

    v[t+1] = v[t];
    v[t+1, choice[t]] = v[t, choice[t]] + lr * pe[t];
  }
}
```

RL – Implementation

cognitive model
statistics
computing

```
model {
  vector[2] v[nTrials+1];
  real pe[nTrials];

  v[1] = initV;

  for (t in 1:nTrials) {
    choice[t] ~ categorical_logit( tau * v[t] );
    pe[t] = reward[t] - v[t,choice[t]];

    v[t+1] = v[t];
    v[t+1, choice[t]] = v[t, choice[t]] + lr * pe[t];
  }
}
```

```
model {
  vector[2] v;
  real pe;

  v = initV;

  for (t in 1:nTrials) {
    choice[t] ~ categorical_logit( tau * v );
    pe = reward[t] - v[choice[t]];

    v[choice[t]] = v[choice[t]] + lr * pe;
  }
}
```

RL – Fitting with Stan

cognitive model
statistics
computing

```
…/06.reinforcement_learning/_scripts/reinforcement_learning_single_parm_main.R
```

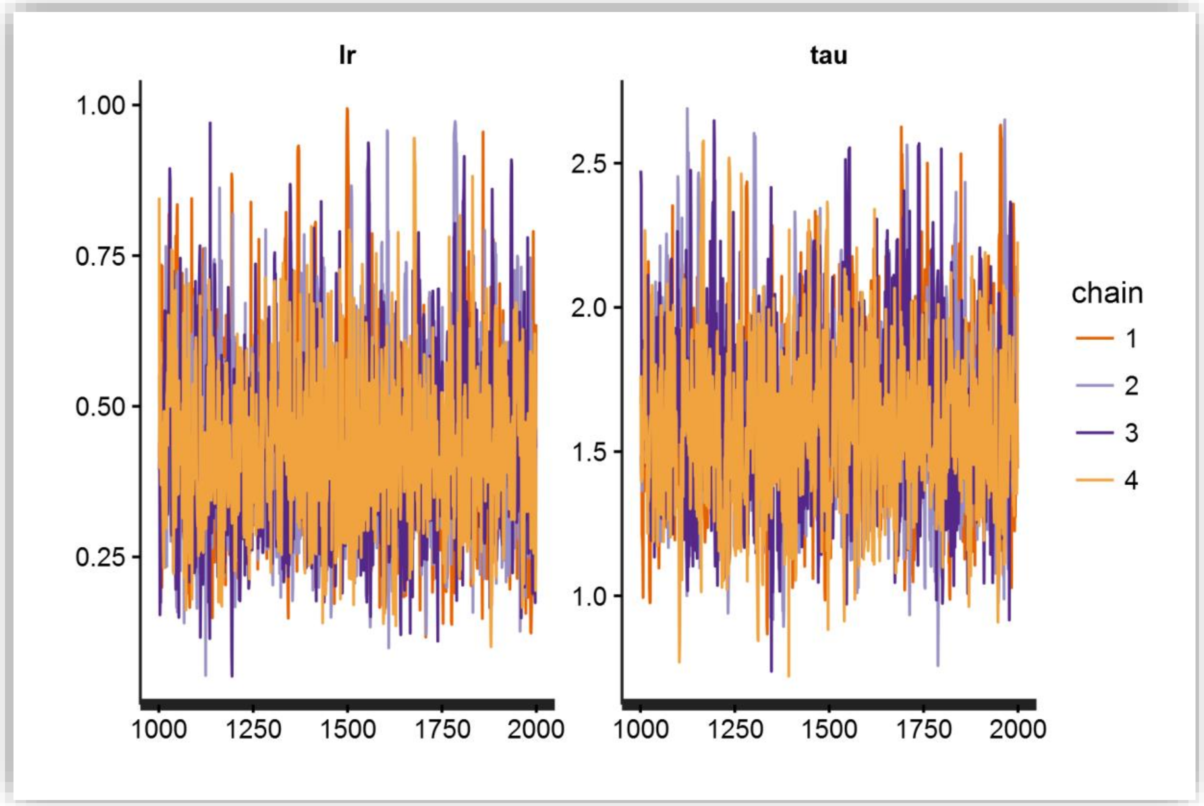
TASK: fit the model for single participants

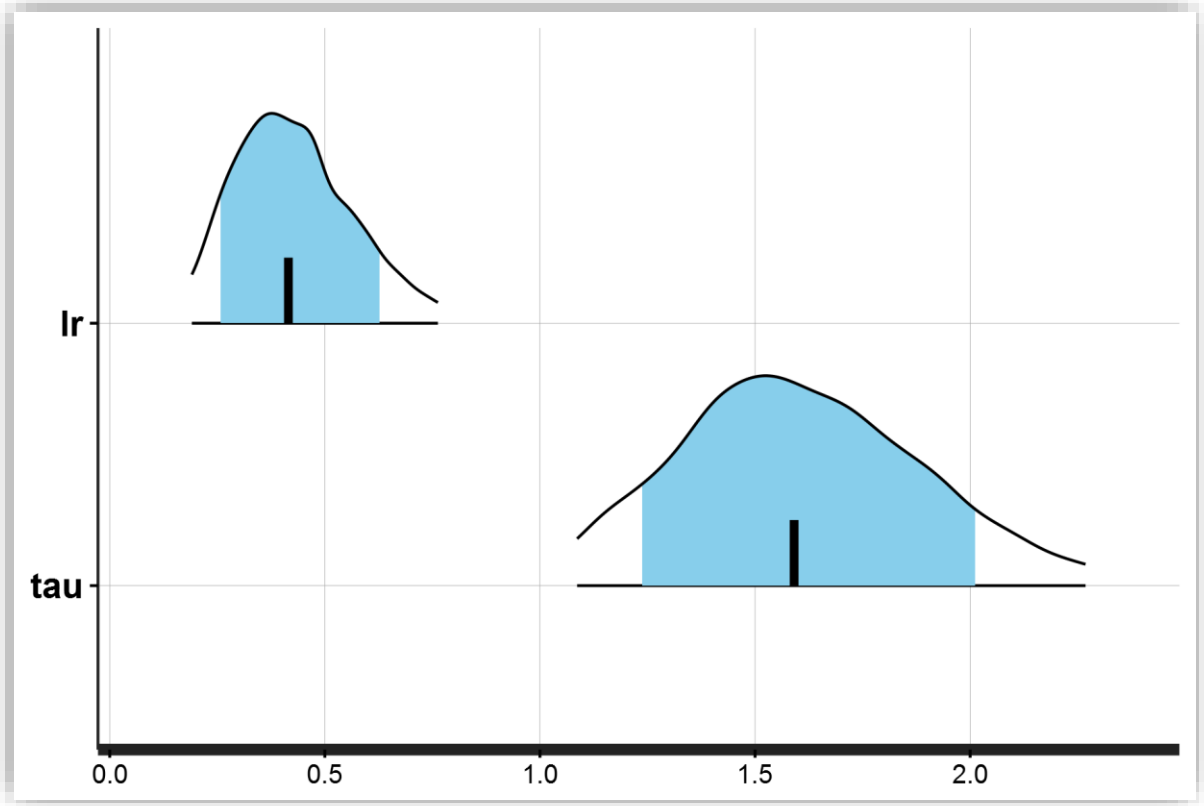
```
> source('_scripts/reinforcement_learning_single_parm_main.R’) # a function

> fit_rl1 <- run_rl_sp(multiSubj = FALSE)
```

```
> load('_data/rl_sp_ss.RData')
> head(rl_ss)
     [,1] [,2]
[1,]    2   -1
[2,]    1    1
[3,]    1    1
[4,]    1    1
[5,]    2   -1
[6,]    1    1
```

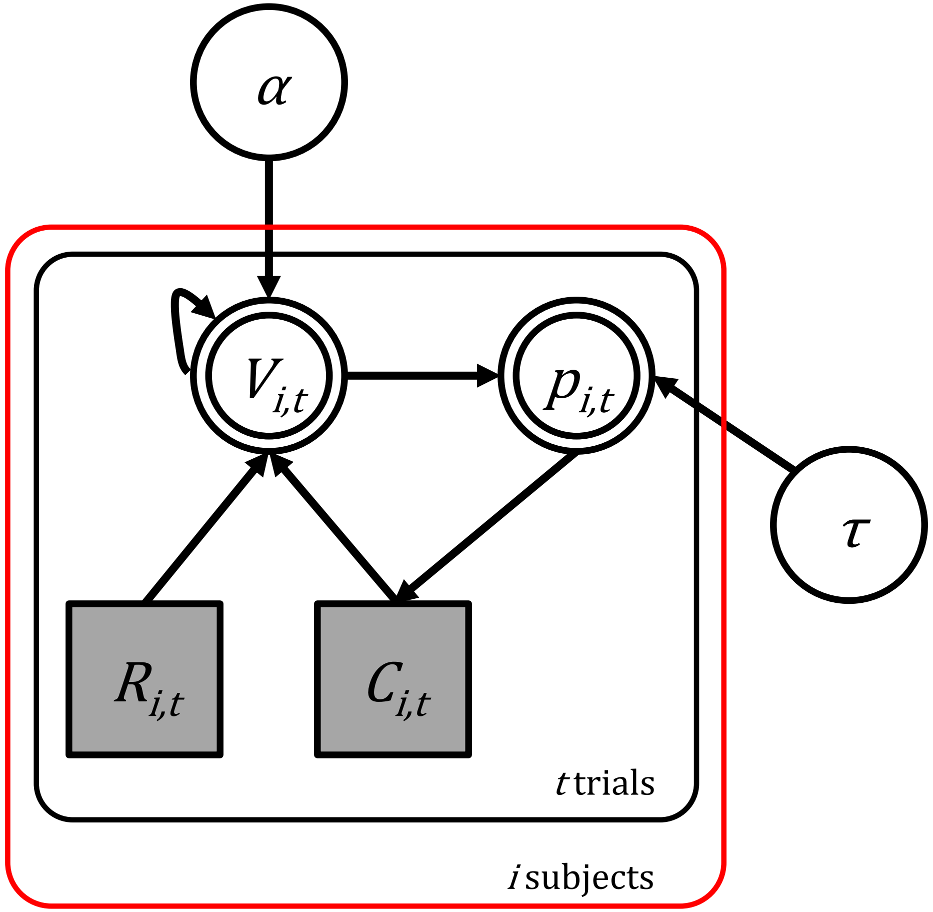
RL – MCMC Output

cognitive model
statistics
computing





Fitting Multiple Participants as ONE




```
model {
  for (s in 1:nSubjects) {
    vector[2] v;
    real pe;
    v = initV;

    for (t in 1:nTrials) {
      choice[s,t] ~ categorical_logit( tau * v );
      pe = reward[s,t] - v[choice[s,t]];
      v[choice[s,t]] = v[choice[s,t]] + lr * pe;
    }
  }
}
```

Exercise X

cognitive model
statistics
computing

```
…/06.reinforcement_learning/_scripts/reinforcement_learning_single_parm_main.R
```

TASK:

(1) complete the model (Tip: the for-loop)

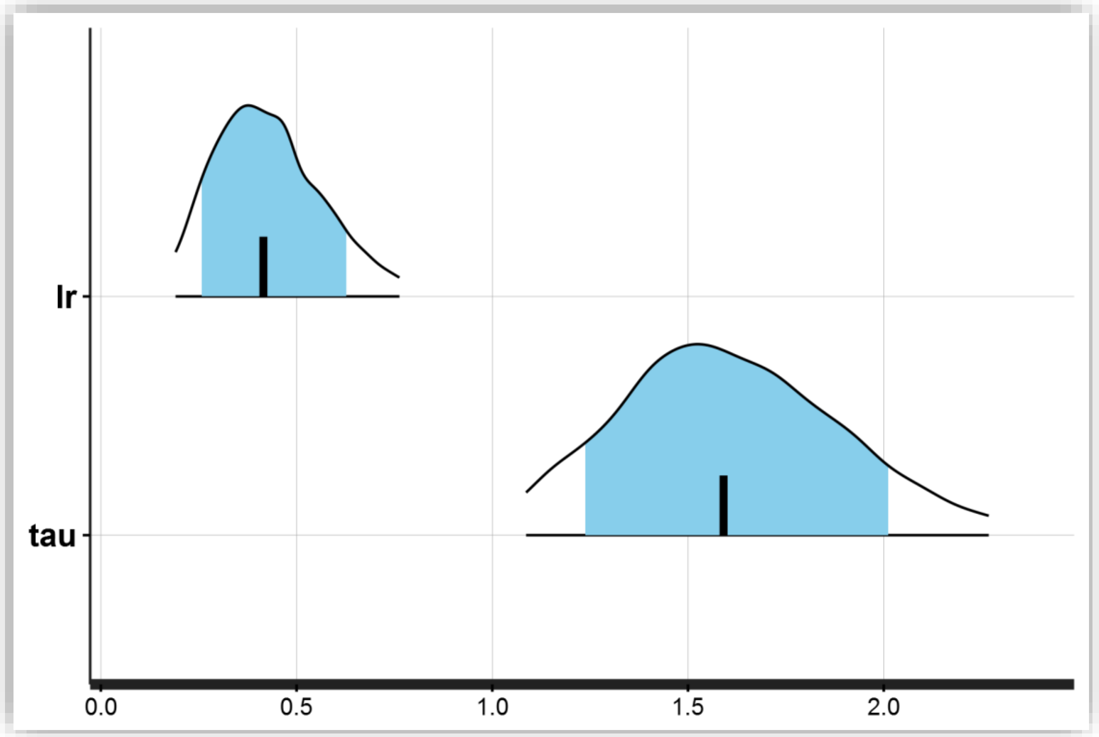
(2) fit the model for multiple participants (assuming same parameters)

```
> source('_scripts/reinforcement_learning_single_parm_main.R')

> fit_rl2 <- run_rl_sp(multiSubj = TRUE)
```

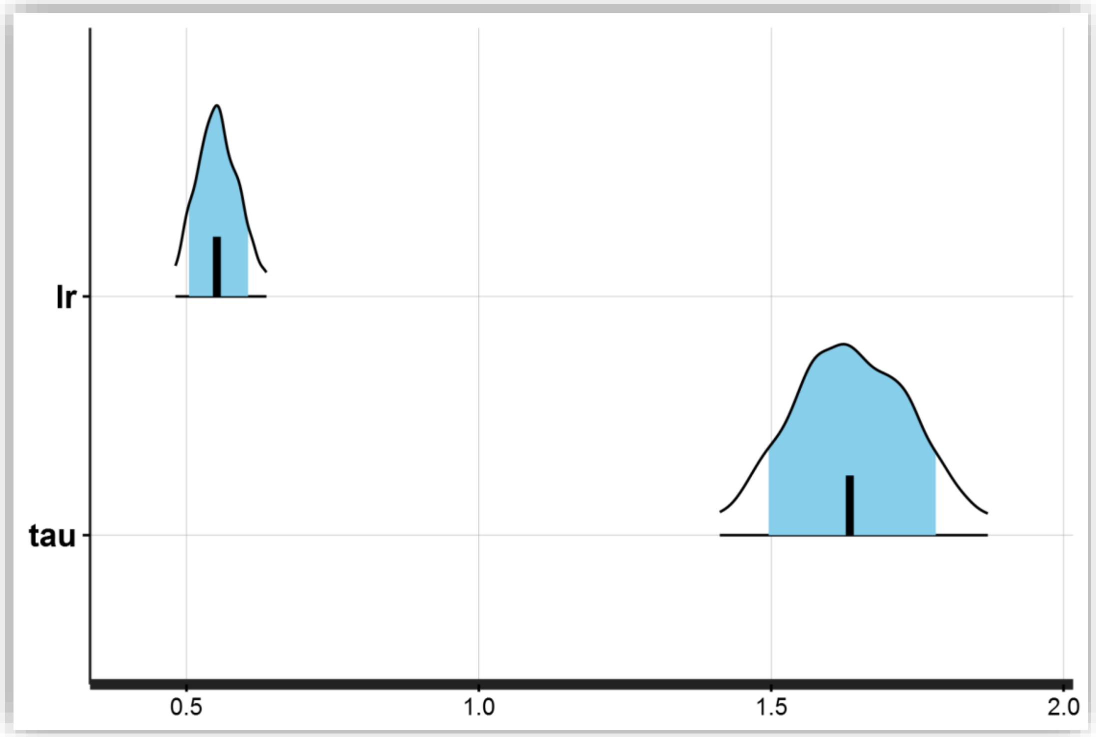
Exercise X

cognitive model
statistics
computing

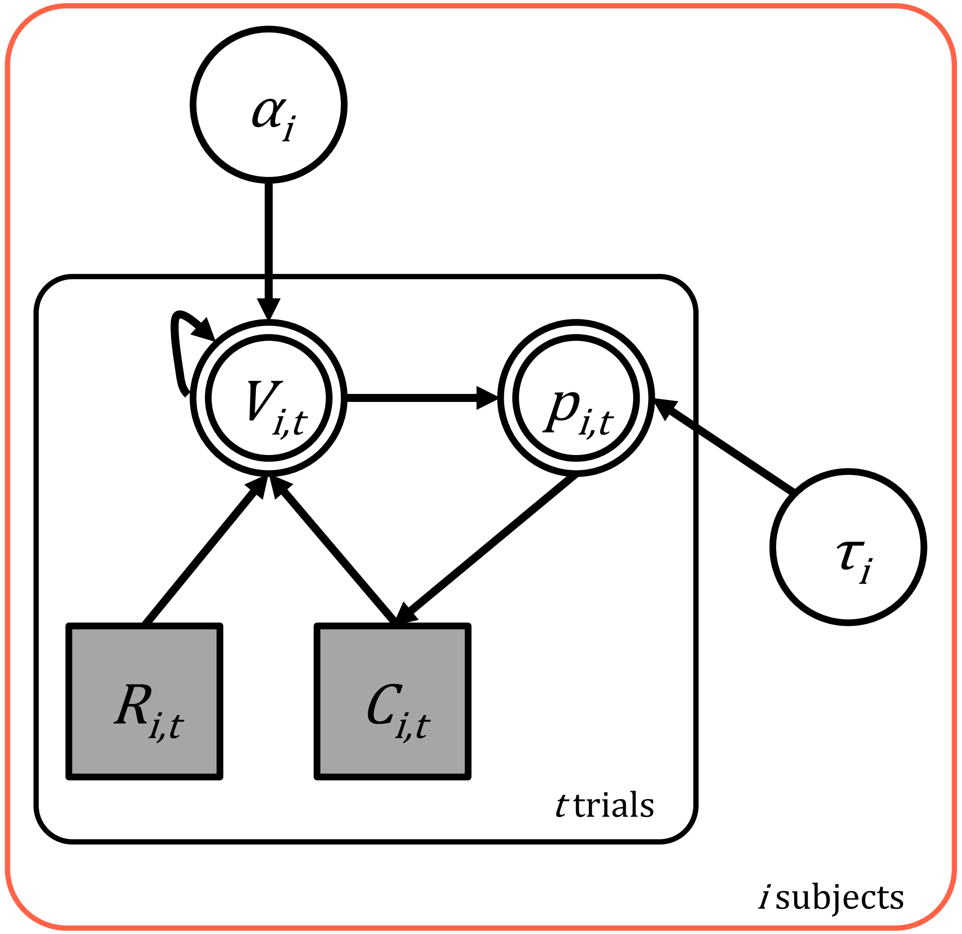
N = 1



N = 10



Fitting Multiple Participants Independently



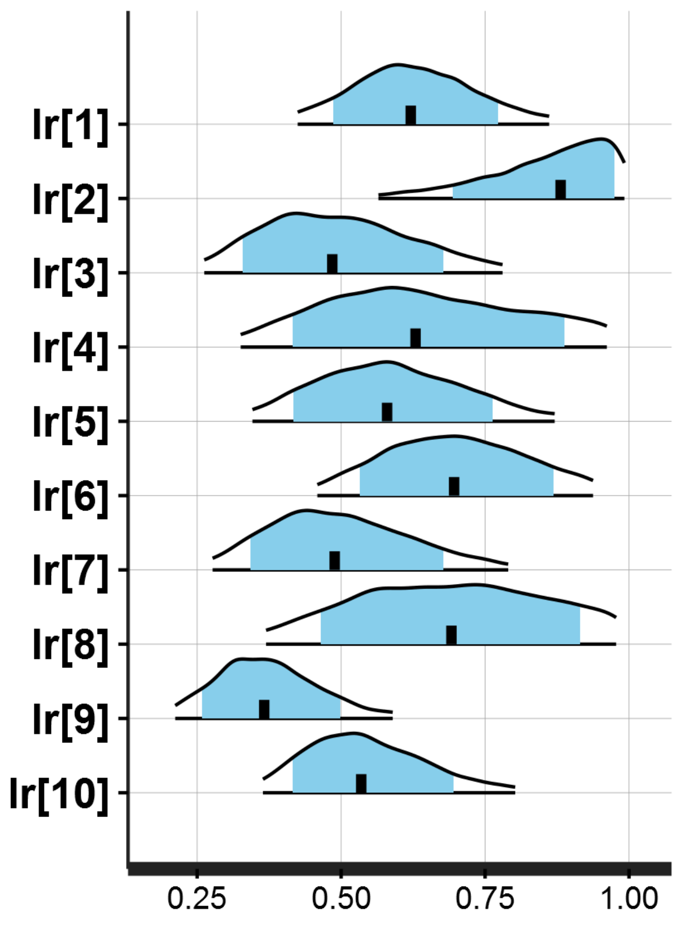

```
model {
  for (s in 1:nSubjects) {
    vector[2] v;
    real pe;
    v = initV;

    for (t in 1:nTrials) {
      choice[s,t] ~ categorical_logit( tau[s] * v );
      pe = reward[s,t] - v[choice[s,t]];
      v[choice[s,t]] = v[choice[s,t]] + lr[s] * pe;
    }
  }
}
```

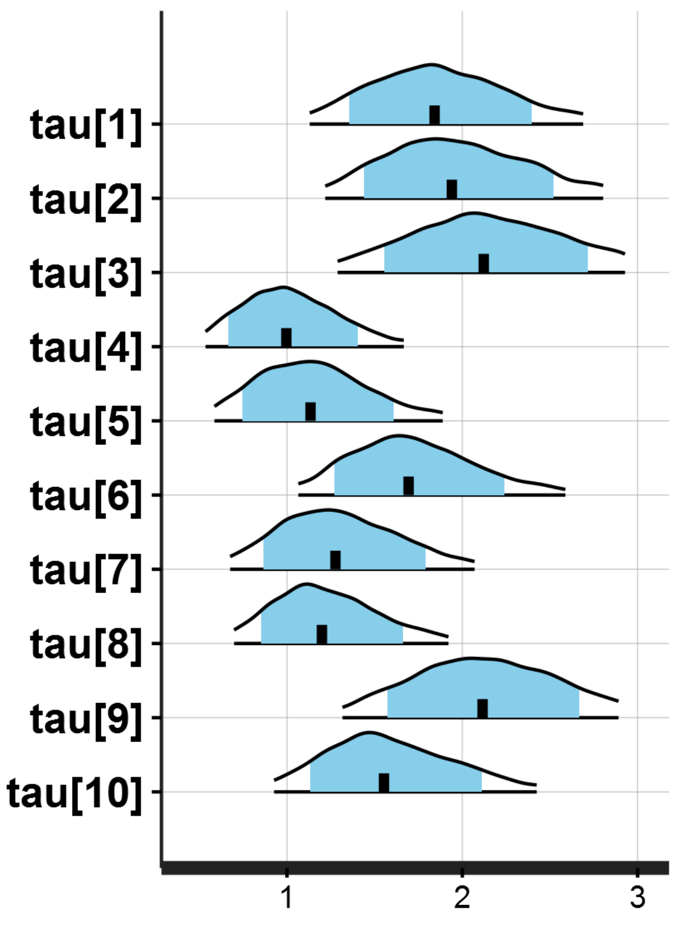
Individual Fitting

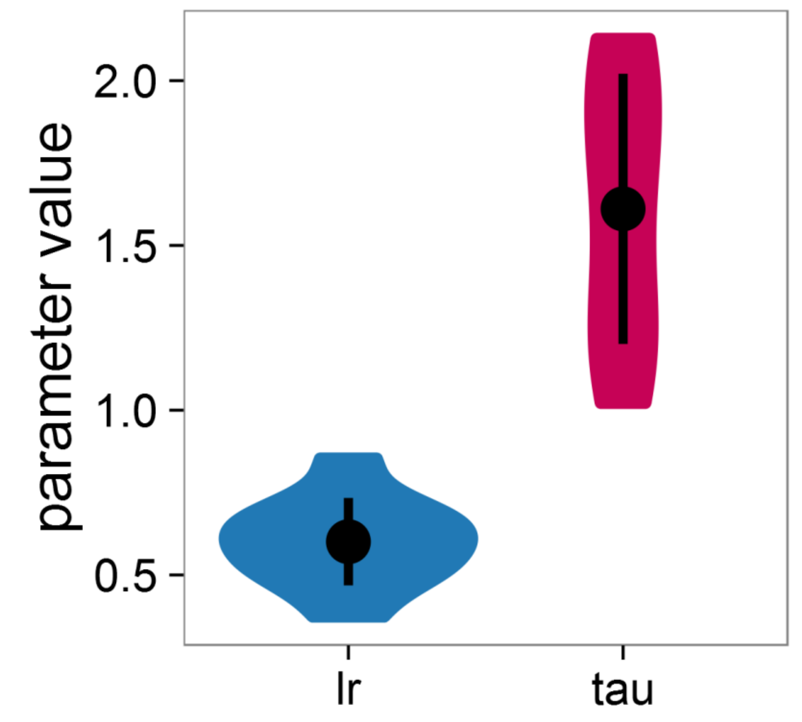
cognitive model

statistics

computing



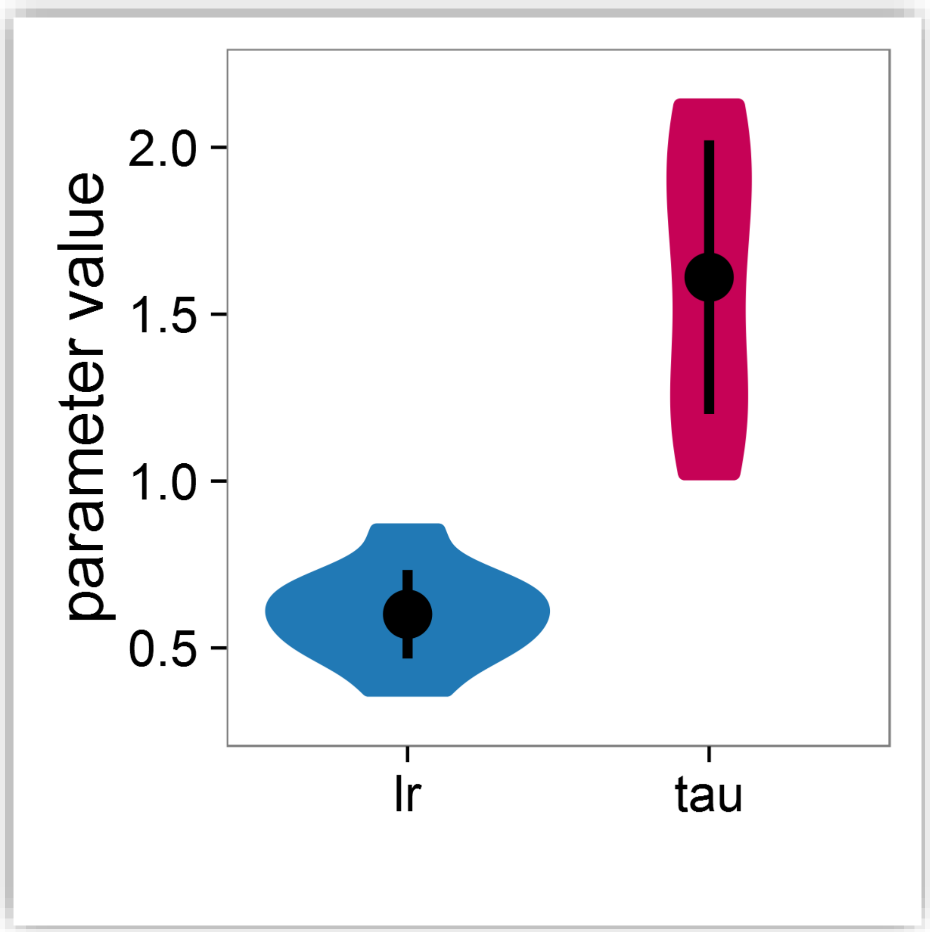




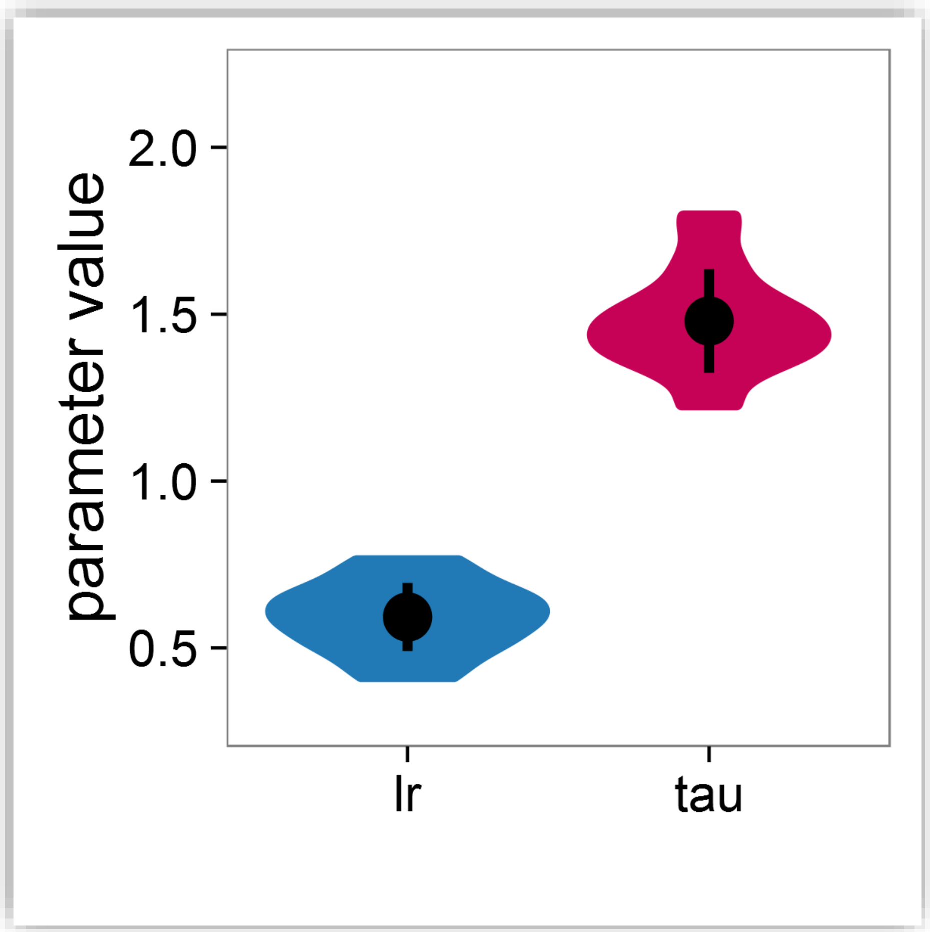
Comparing with True Parameters

cognitive model
statistics
computing

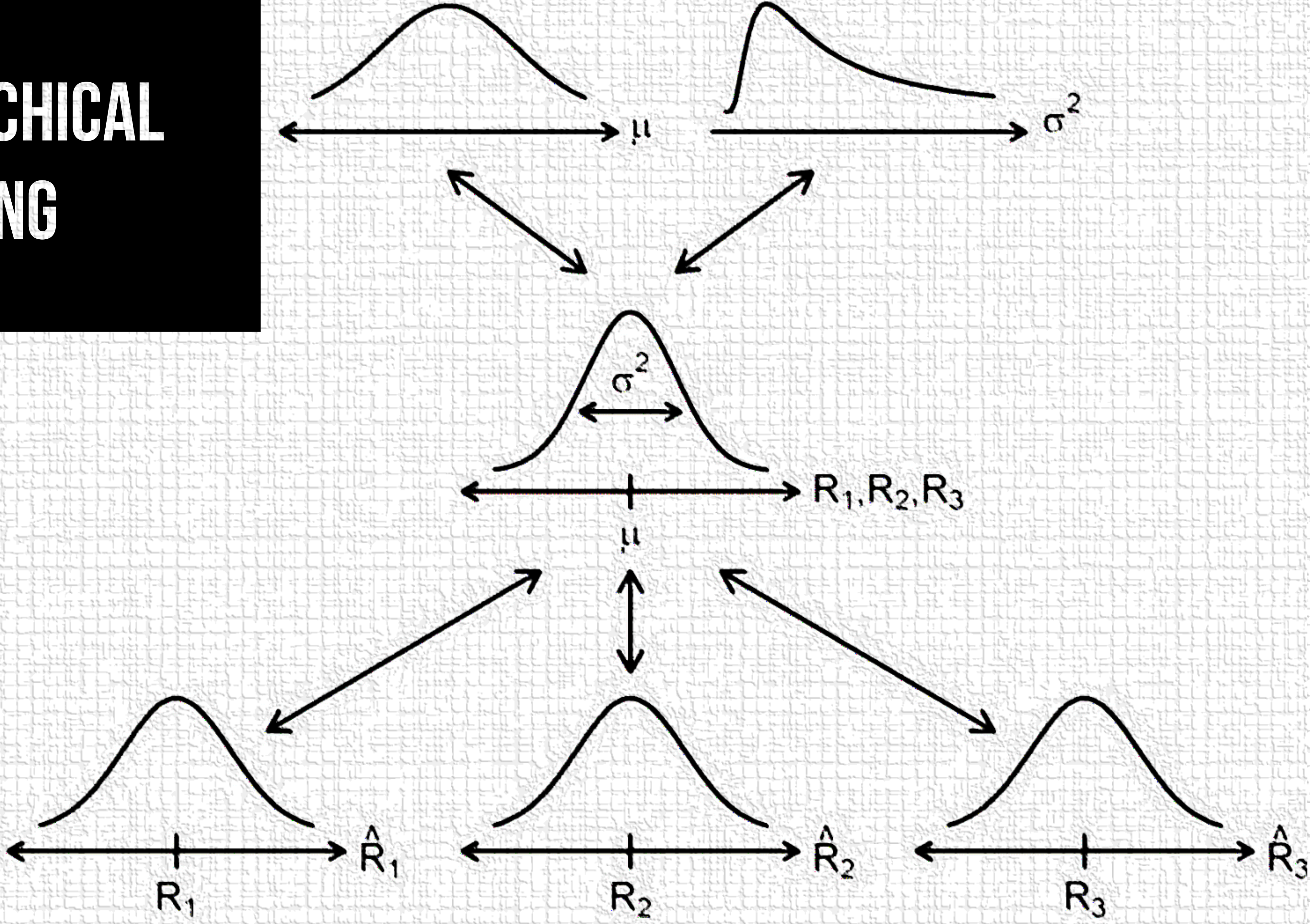
Posterior Means

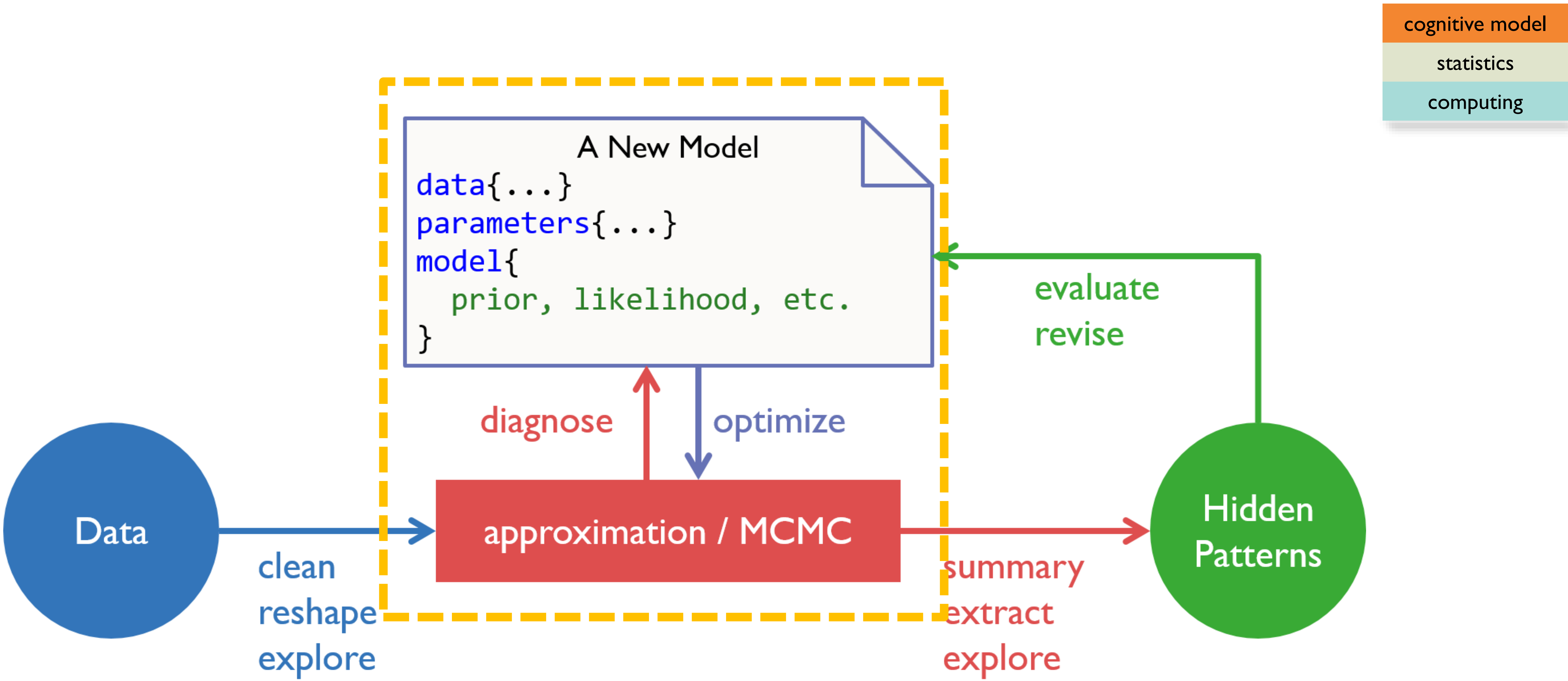


True Parameters

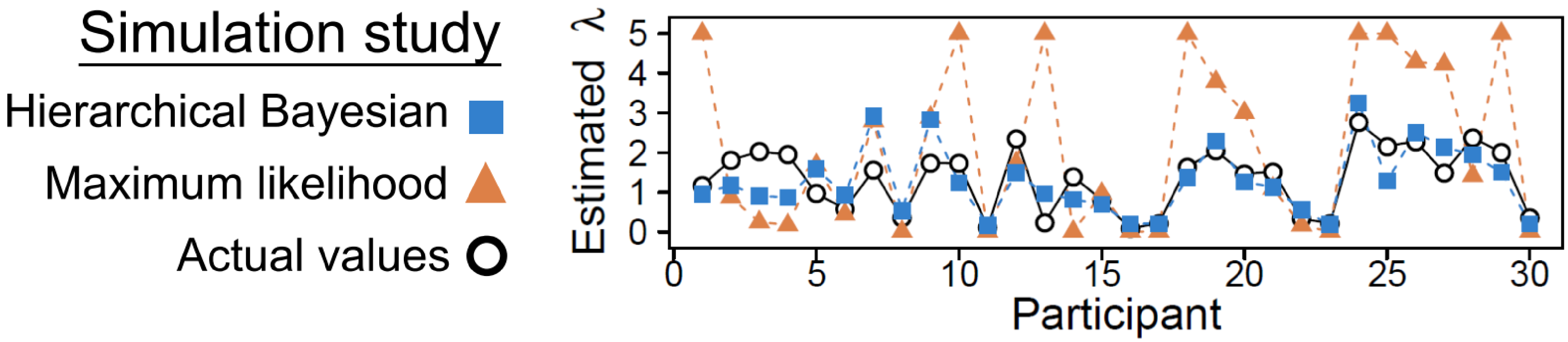


HIERARCHICAL MODELING




cognitive model
statistics
computing
A New Model
data{...}
parameters{...}
model{
prior, likelihood, etc.
}
evaluate
revise
diagnose
optimize
Data
approximation / MCMC
Hidden Patterns
clean
reshape
explore
summary
extract
explore

Why Hierarchical Bayesian Cognitive Modeling?



Ahn et al. (2011)

Why Hierarchical Bayesian Cognitive Modeling?

Fixed effects

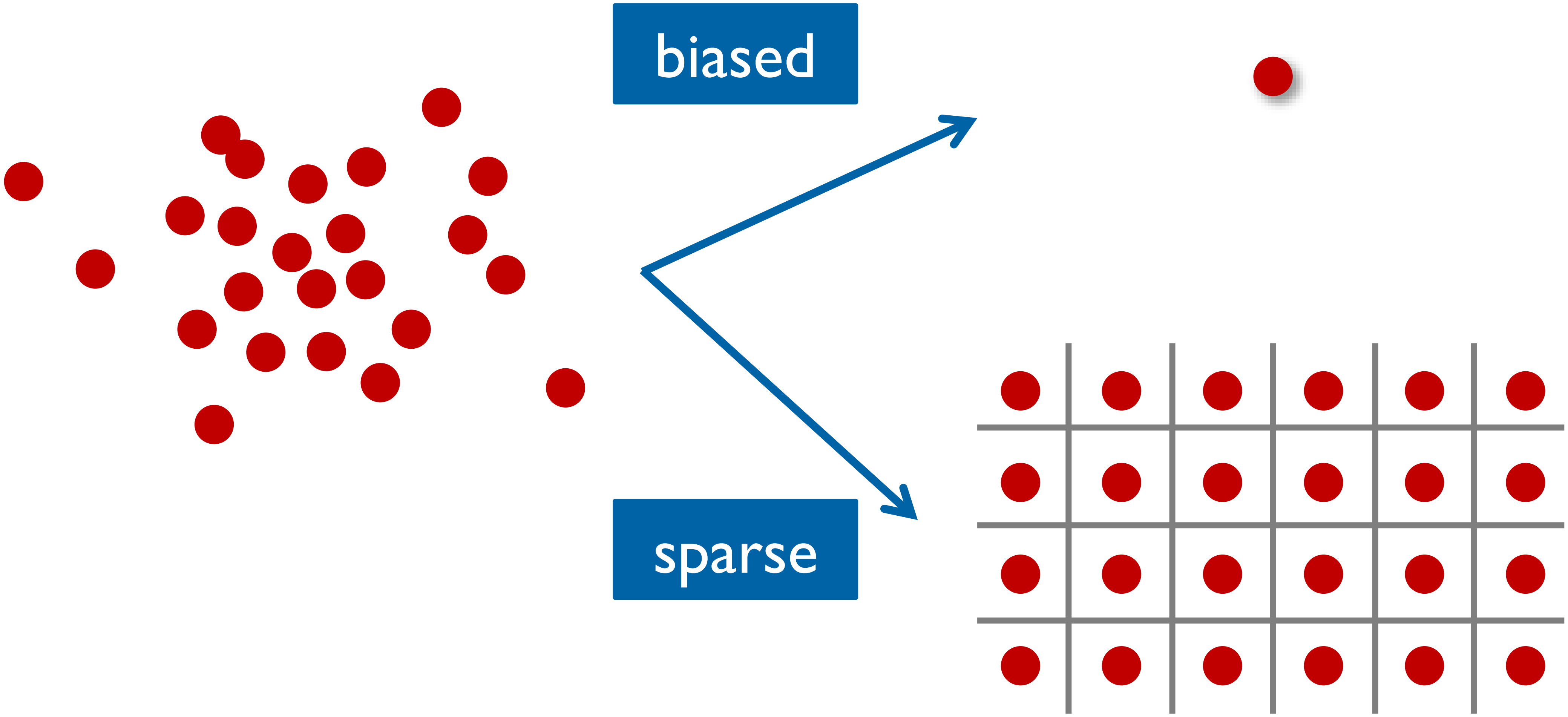
- all subjects are fitted with the same set of parameters
- worse model fit than "random effects"

Random effects

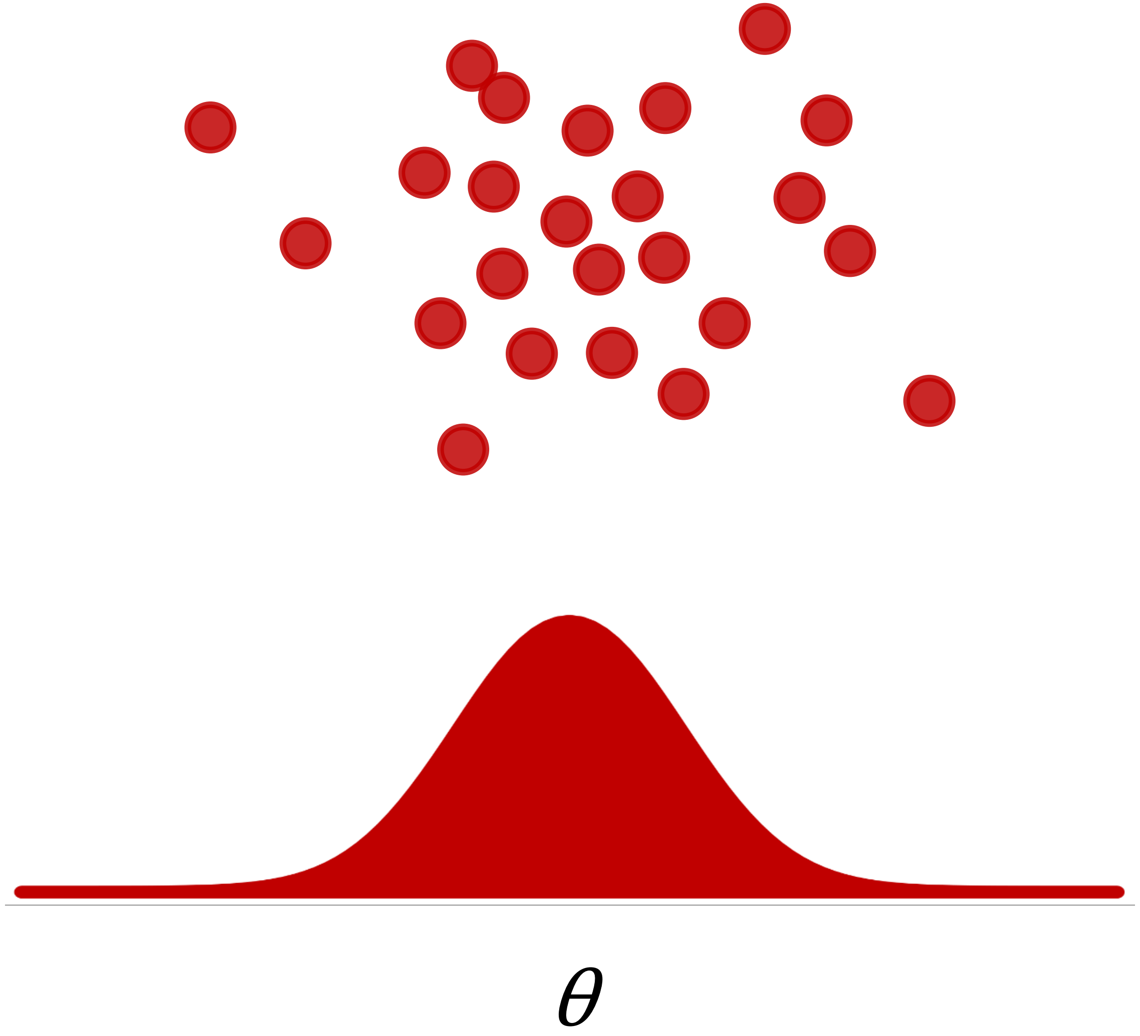
- each subject is fitted independently of the others
- best model fit for each subject
- parameter estimates can be noisy

Adapted from Jan Gläscher's workshop

Fitting Multiple Participants

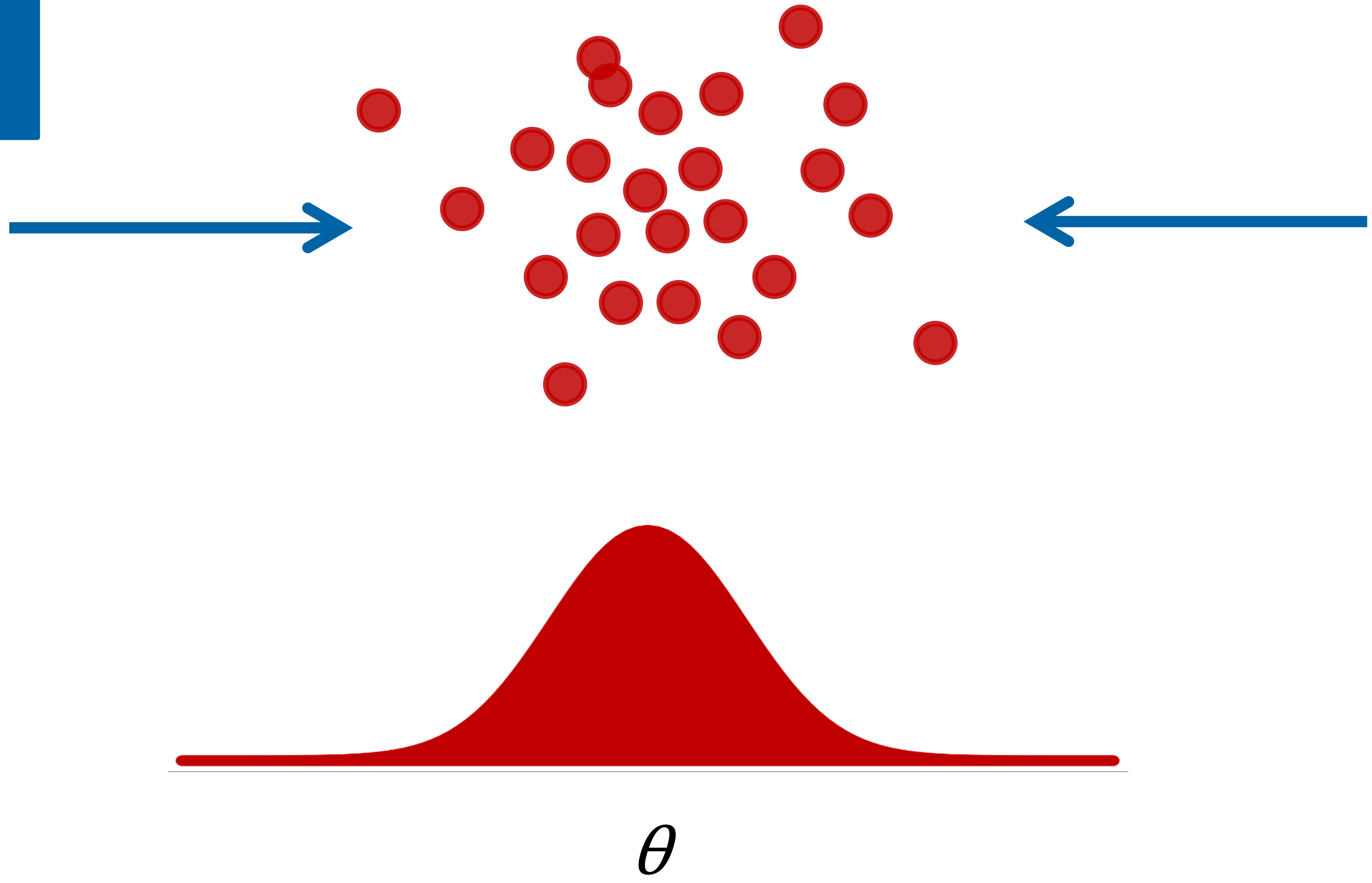


Fitting Multiple Participants



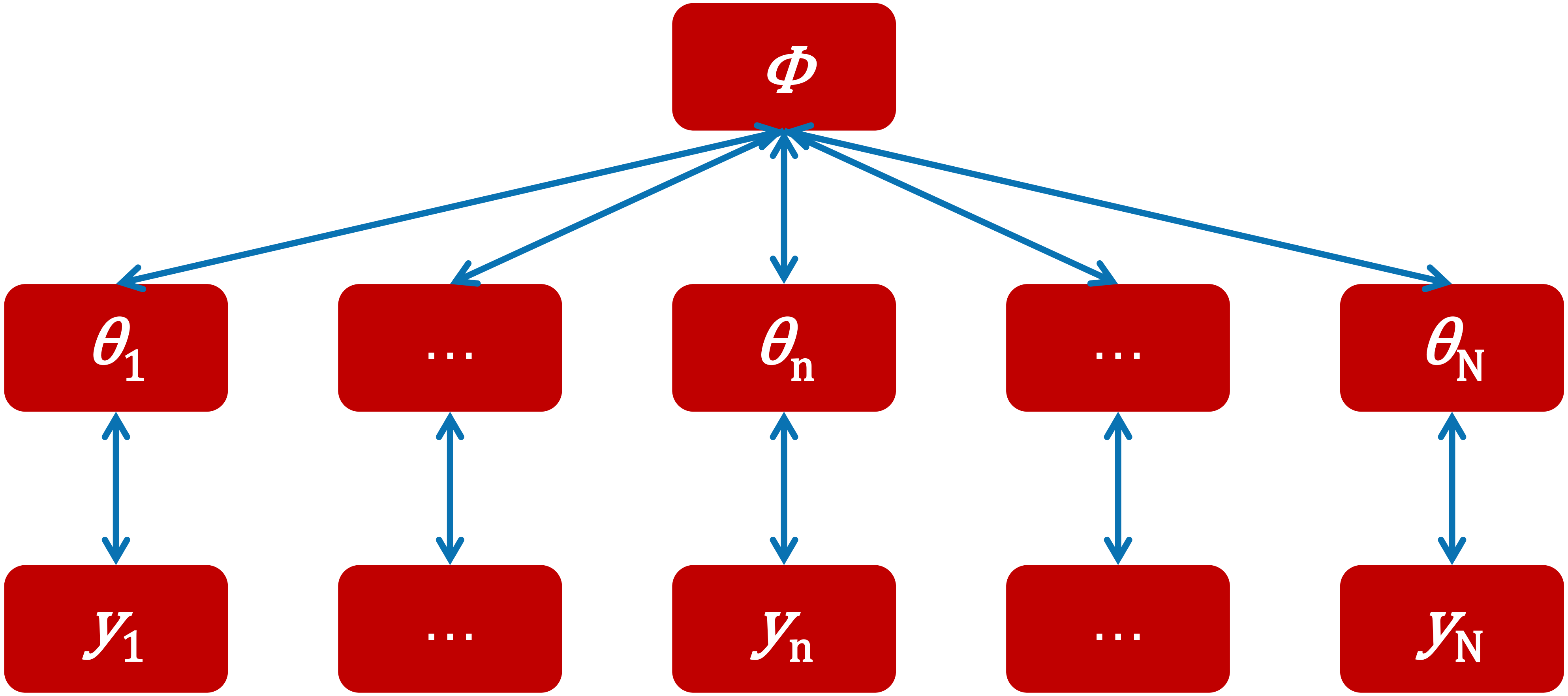
Fitting Multiple Participants

shrinkage effect



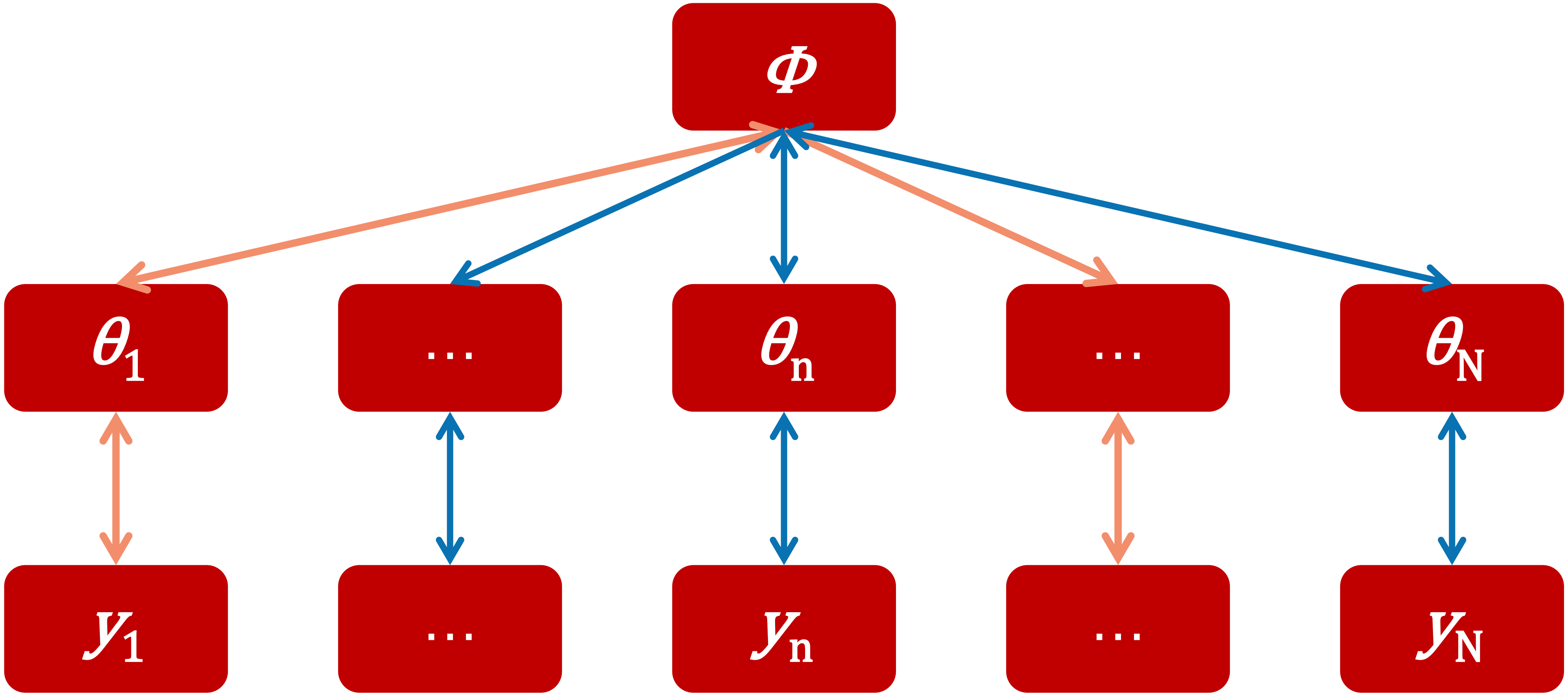
Hierarchical Structure

cognitive model
statistics
computing



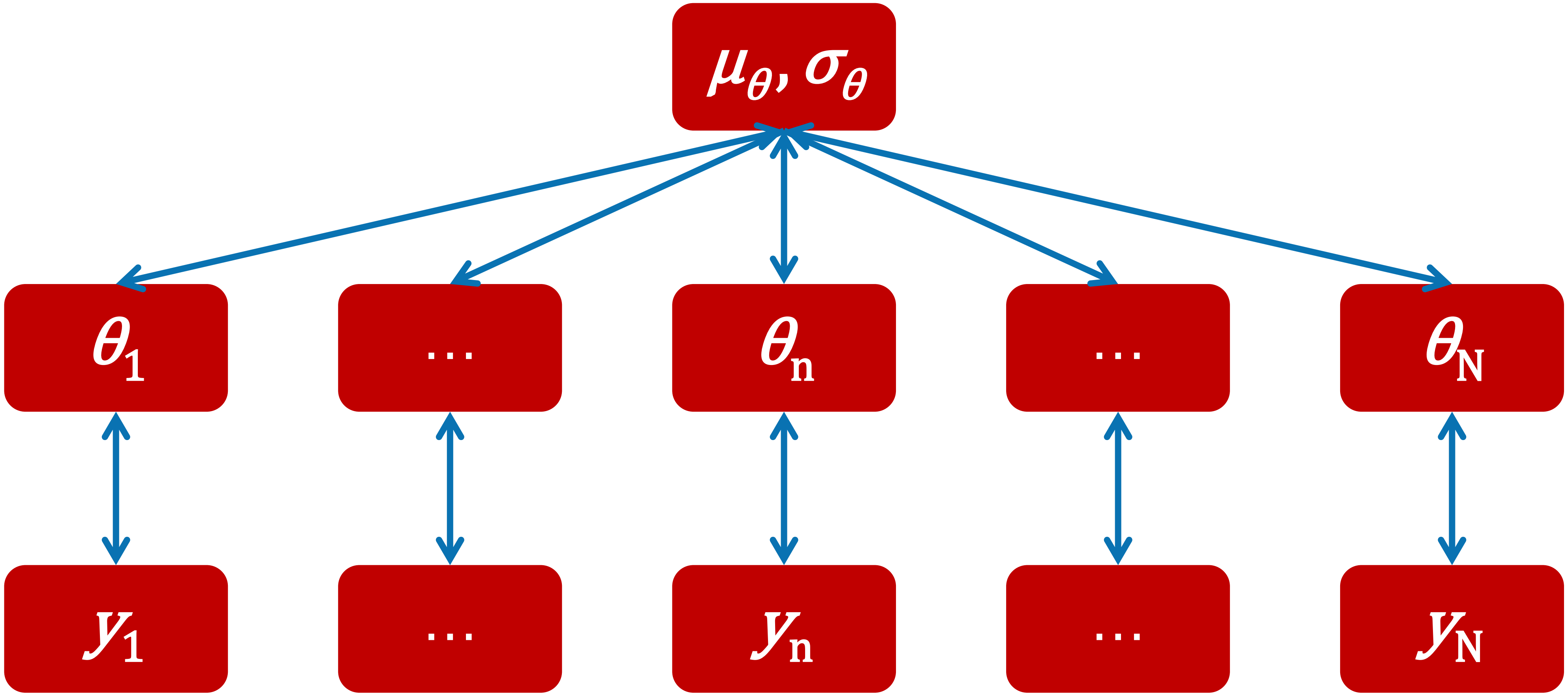
Hierarchical Structure

cognitive model
statistics
computing



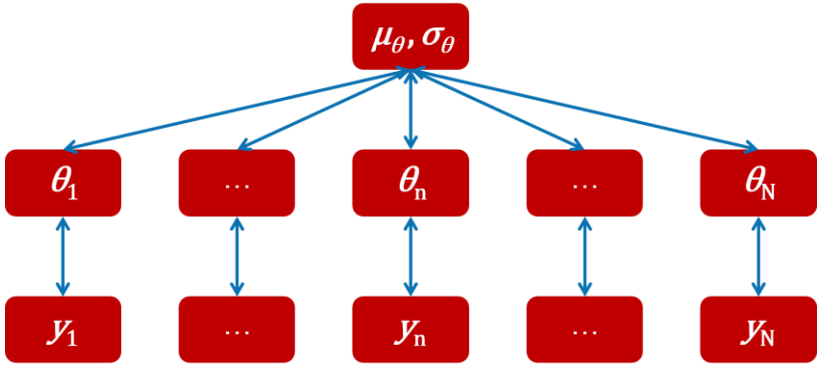
Hierarchical Structure

cognitive model
statistics
computing



cognitive model
statistics
computing

Hierarchical Structure



$$P(\Theta,\Phi \mid D) = \frac{P(D \mid \Theta,\Phi)P(\Theta,\Phi)}{P(D)} \propto P(D \mid \Theta)P(\Theta \mid \Phi)P(\Phi)$$