Reinforcement learning crash course

Quentin Huys

Translational Neuromodeling Unit, University of Zurich and ETH Zurich
University Hospital of Psychiatry Zurich

Computational Psychiatry Course
Zurich, 1.9.2016

Overview

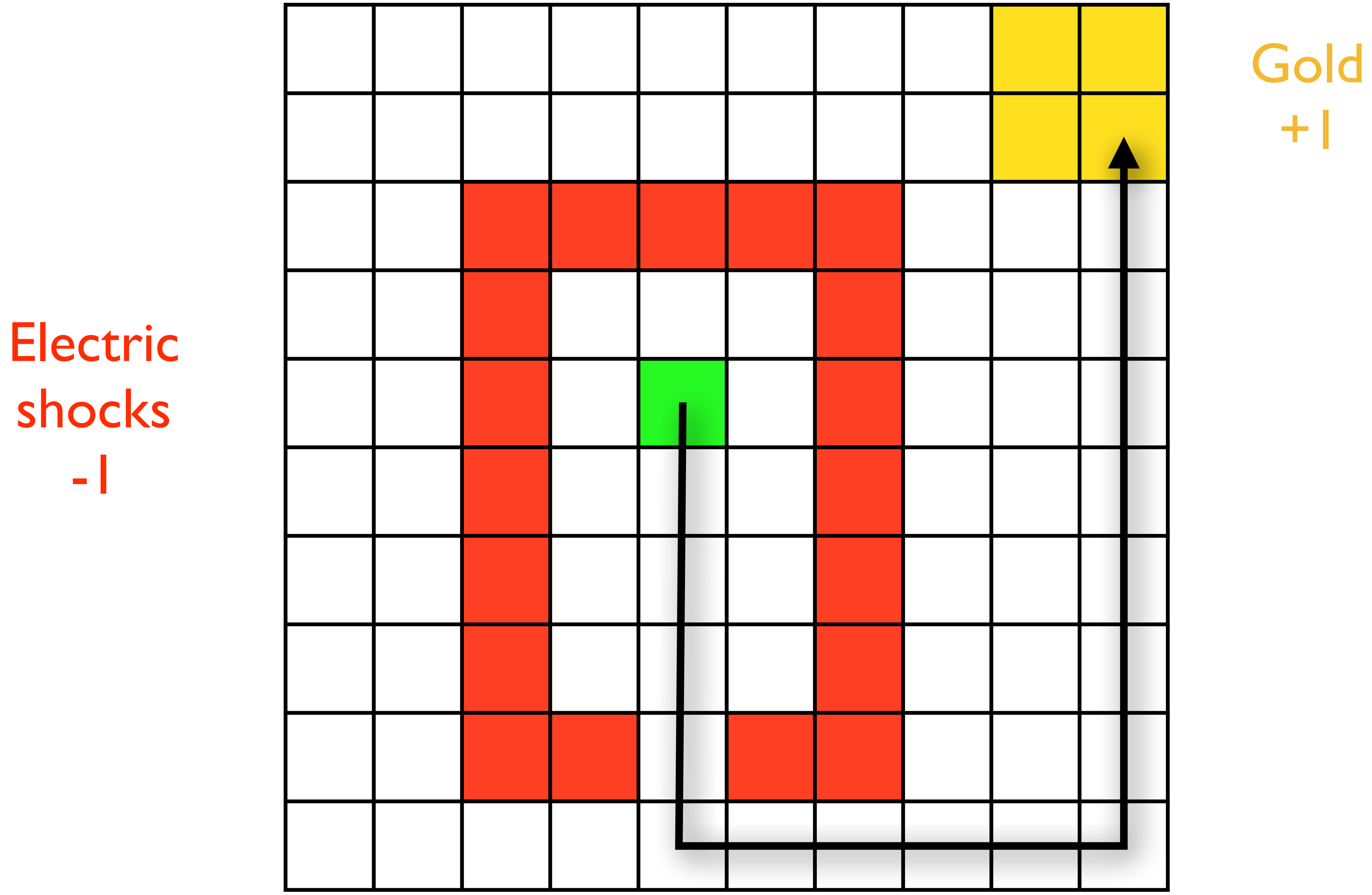
- Reinforcement learning: rough overview
  - mainly following Sutton & Barto 1998
- Dopamine
  - prediction errors and more
- Fitting behaviour with RL models
  - hierarchical approaches

Setup

$$\{a_t\} \leftarrow \underset{\{a_t\}}{\operatorname{argmax}} \sum_{t=1}^{\infty} r_t$$

After Sutton and Barto 1998

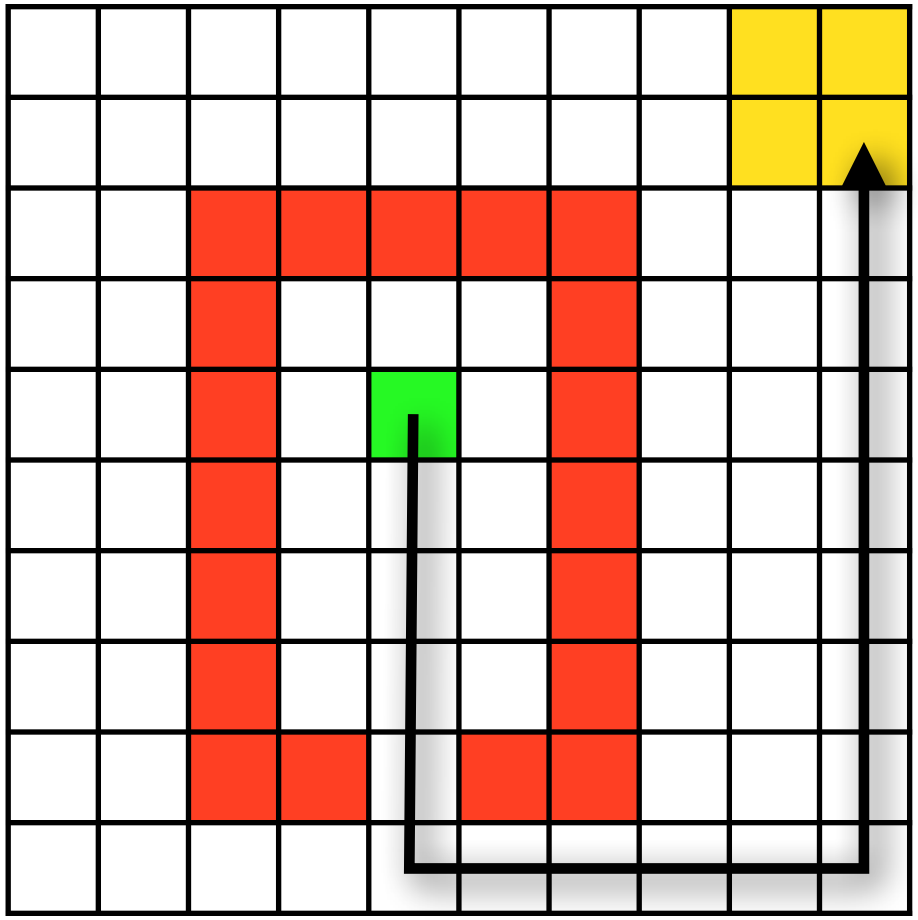
State space



A Markov Decision Problem

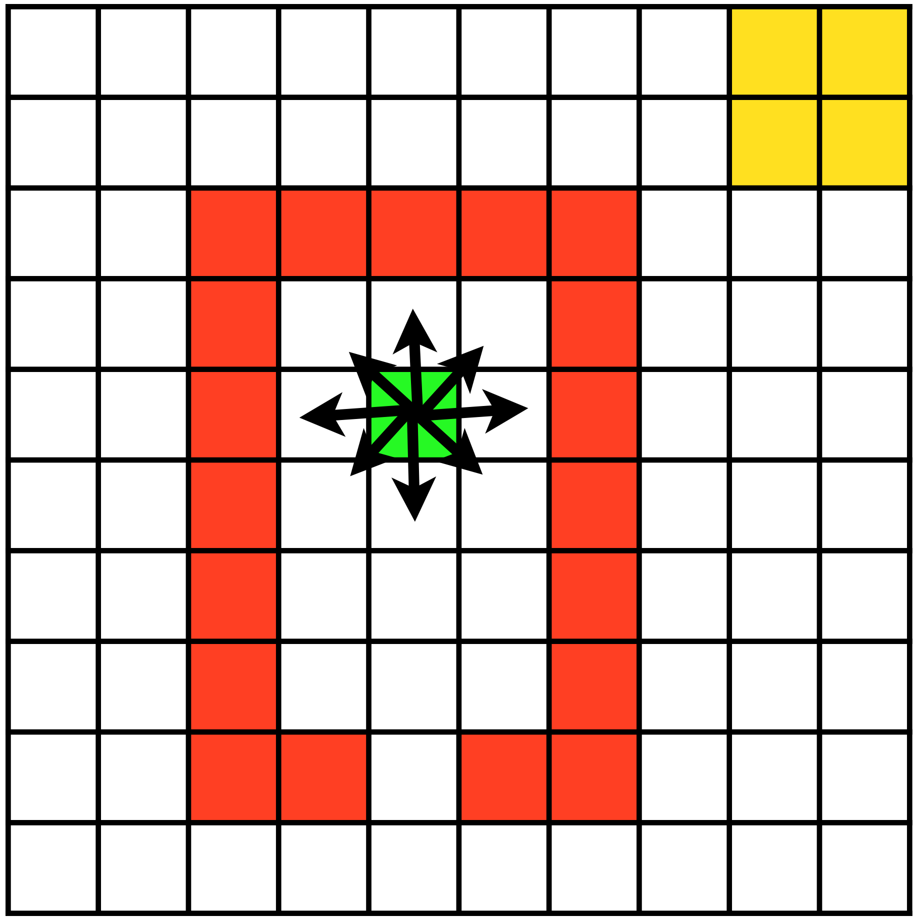
$$\boxed{s_t \in \mathcal{S}}$$

$$\begin{aligned} a_t &\in \mathcal{A} \\ \mathcal{T}^a_{ss'} &= p(s_{t+1}|s_t, a_t) \\ r_t &\sim \mathcal{R}(s_{t+1}, a_t, s_t) \\ \pi(a|s) &= p(a|s) \end{aligned}$$

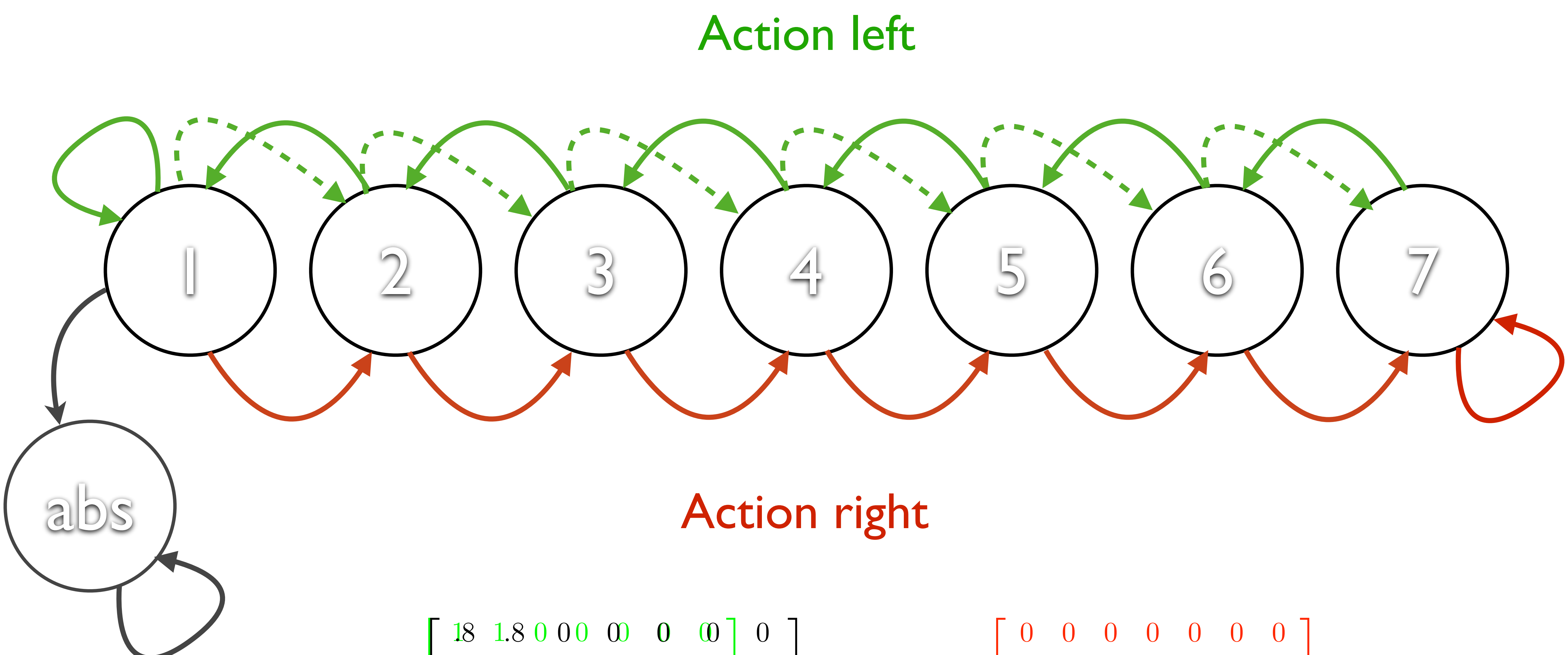


A Markov Decision Problem

$$\begin{aligned} s_t &\in \mathcal{S} \\ \boxed{\begin{aligned} a_t &\in \mathcal{A} \\ \mathcal{T}^a_{ss'} &= p(s_{t+1}|s_t, a_t) \end{aligned}} \\ r_t &\sim \mathcal{R}(s_{t+1}, a_t, s_t) \\ \pi(a|s) &= p(a|s) \end{aligned}$$



Actions



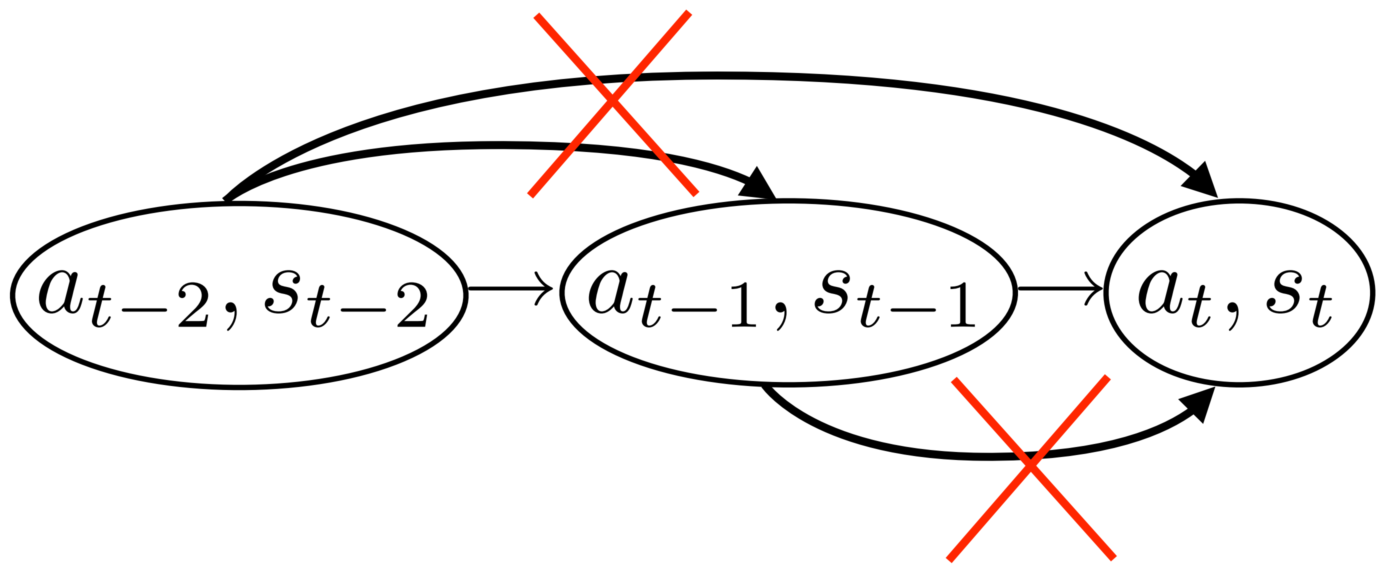
$$T^{\text{left}} = \begin{bmatrix} .8 & .8 & 0 & 0 & 0 & 0 & 0 \\ .2 & .2 & .8 & 0 & 0 & 0 & 0 \\ 0 & 0 & .2 & .8 & 0 & 0 & 0 \\ 0 & 0 & 0 & .2 & .8 & 0 & 0 \\ 0 & 0 & 0 & 0 & .2 & .8 & 0 \\ 0 & 0 & 0 & 0 & 0 & .2 & .8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad T^{\text{right}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Noisy: plants, environments, agent

Absorbing state -> max eigenvalue < 1

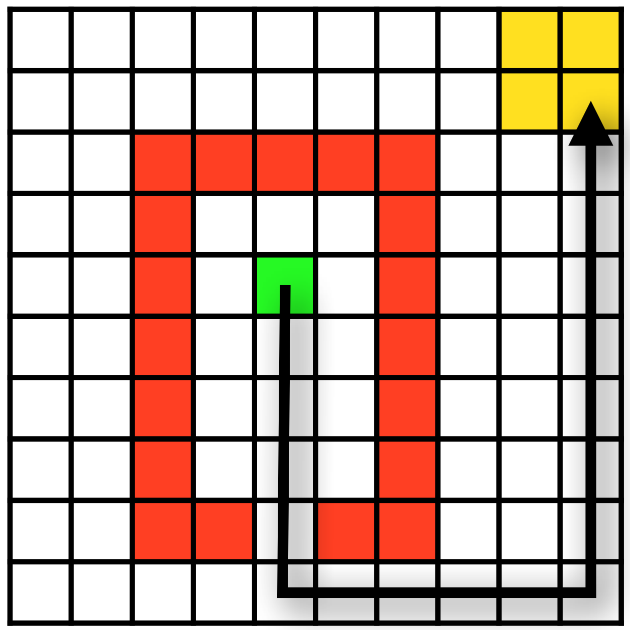
Markovian dynamics

$$p(s_{t+1}|a_t, s_t, a_{t-1}, s_{t-1}, a_{t-2}, s_{t-2}, \cdots) = p(s_{t+1}|a_t, s_t)$$



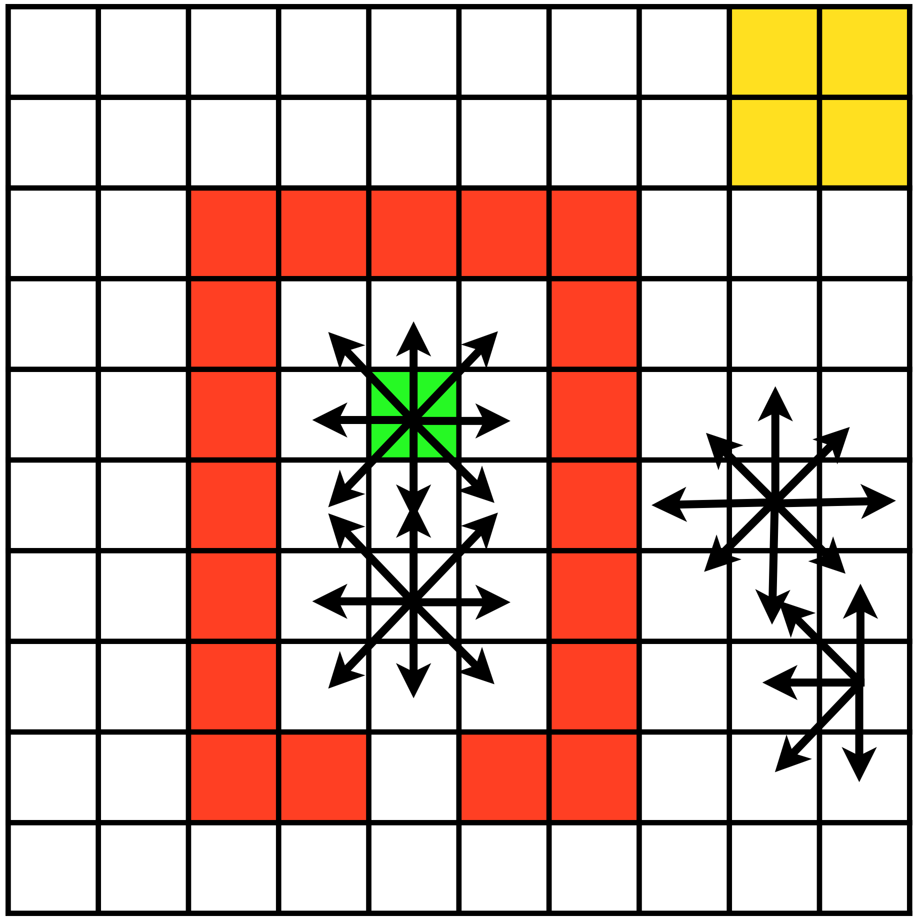
Velocity

$$s' = [\text{position}] \rightarrow s' = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$



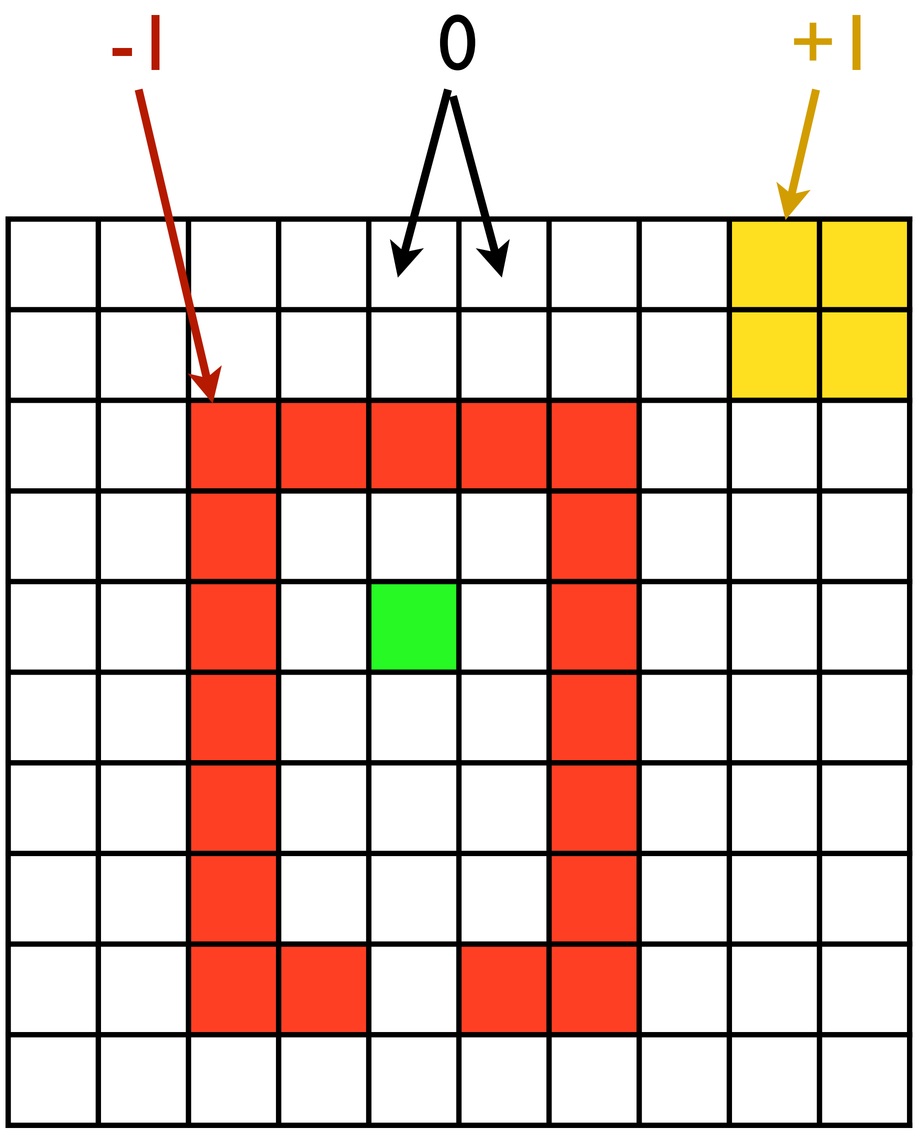
A Markov Decision Problem

$$\begin{aligned} s_t &\in \mathcal{S} \\ a_t &\in \mathcal{A} \\ \mathcal{T}^a_{ss'} &= p(s_{t+1}|s_t, a_t) \\ r_t &\sim \mathcal{R}(s_{t+1}, a_t, s_t) \\ \pi(a|s) &= p(a|s) \end{aligned}$$



A Markov Decision Problem

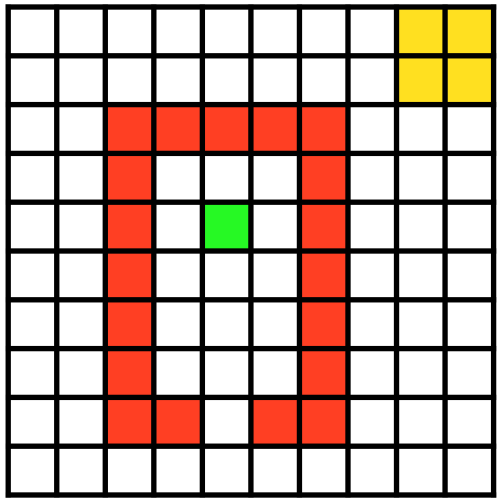
$$\begin{aligned} s_t &\in \mathcal{S} \\ a_t &\in \mathcal{A} \\ \mathcal{T}^a_{ss'} &= p(s_{t+1}|s_t, a_t) \\ \boxed{r_t \sim \mathcal{R}(s_{t+1}, a_t, s_t)} \\ \pi(a|s) &= p(a|s) \end{aligned}$$



Tall orders

- Aim: maximise total future reward

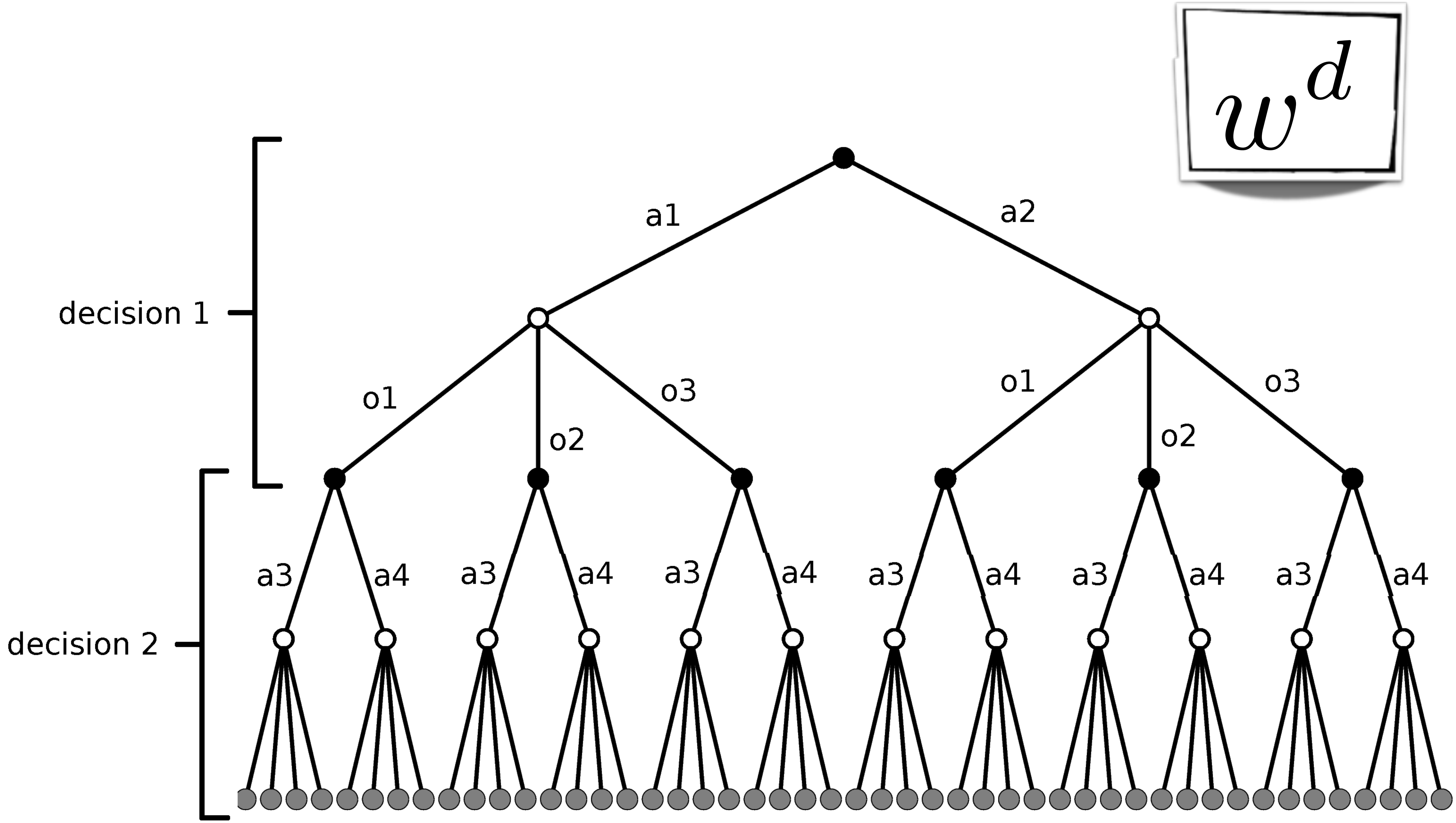
$$\sum_{t=1}^{\infty} r_t$$



- i.e. we have to sum over paths through the future and weigh each by its probability

- Best policy achieves best long-term reward

Exhaustive tree search



Decision tree

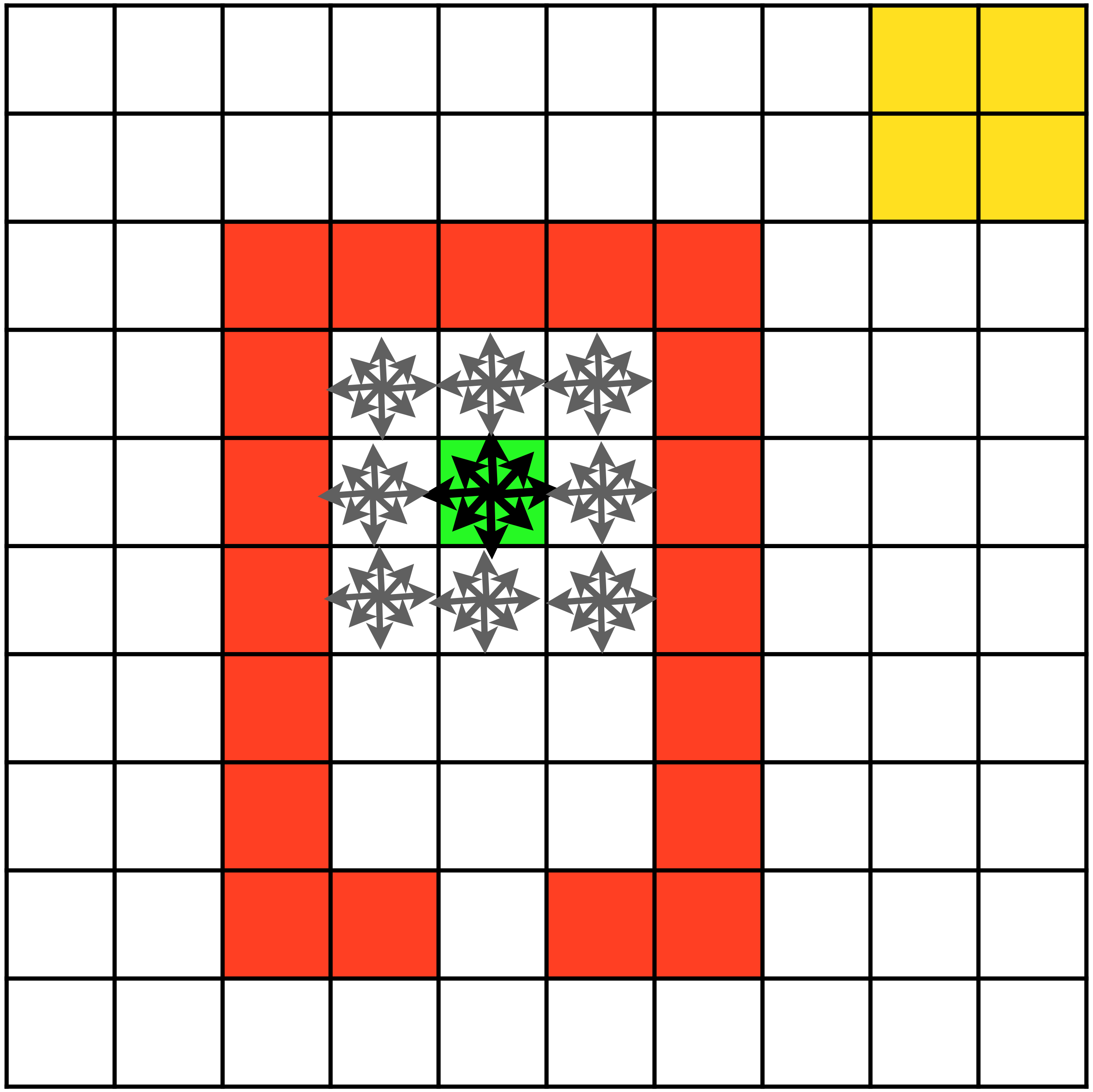
$$\sum_{t=1}^{\infty} r_t$$

8

64

512

...



Policy for this talk

- Pose the problem mathematically
- Policy evaluation
- Policy iteration
- Monte Carlo techniques: experience samples
- TD learning

Policy

Evaluate $\rightleftarrows$ Update

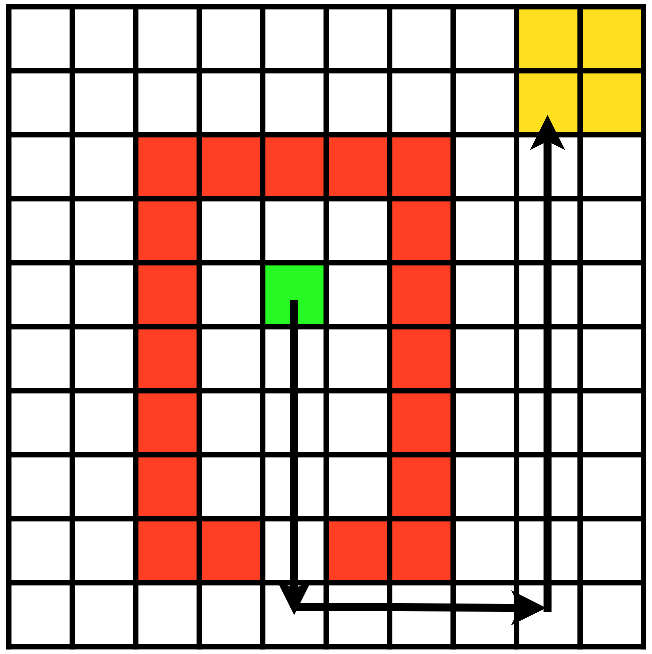
Evaluating a policy

- Aim: maximise total future reward

$$\sum_{t=1}^{\infty} r_t$$

- To know which is best, evaluate it first
- The policy determines the expected reward from each state

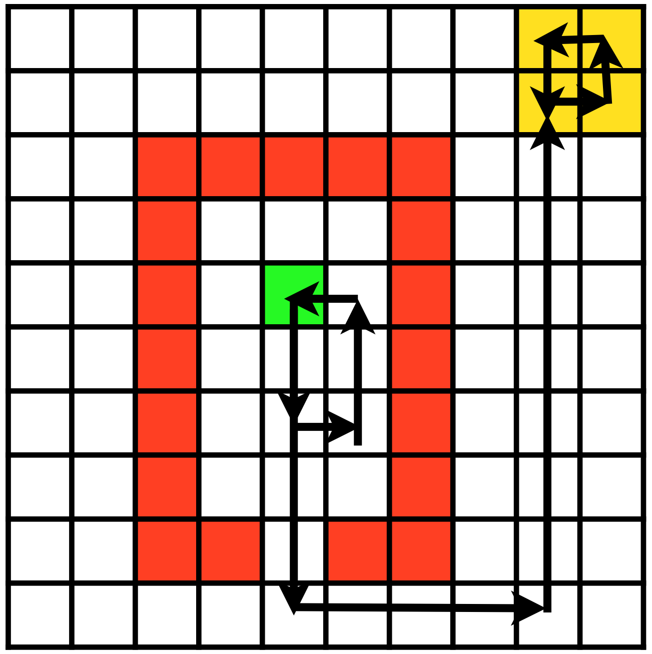
$$\mathcal{V}^{\pi}(s_1) \quad = \quad \mathbb{E}\left[\sum_{t=1}^{\infty} r_t | s_1 = 1, a_t \sim \pi\right]$$



Discounting

- Given a policy, each state has an expected value

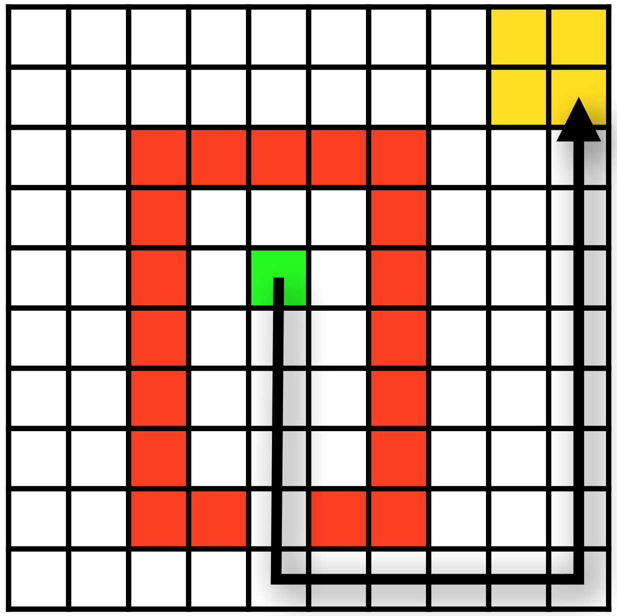
$$\mathcal{V}^{\pi}(s_1) = \mathbb{E}\left[\sum_{t=1}^{\infty} r_t | s_1 = 1, a_t \sim \pi\right]$$



- But: $\sum_{t=0}^{\infty} r_t = \infty$
- Episodic $\sum_{t=0}^{T} r_t < \infty$
- Discounted
  - infinite horizons $\sum_{t=0}^{\infty} \gamma^t r_t < \infty$
  - finite, exponentially distributed horizons

$$\sum_{t=0}^{T} \gamma^t r_t \qquad T \sim \frac{1}{\tau} e^{t/\tau}$$

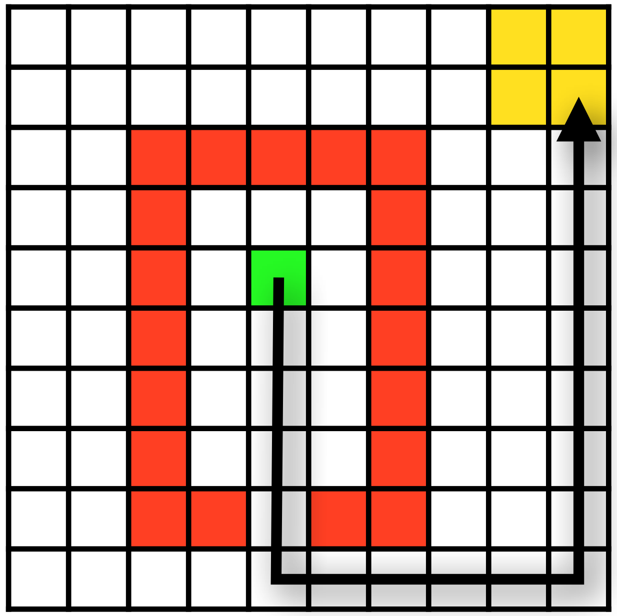
Markov Decision Problems



$$
\begin{aligned}
V^{\pi}(s_t) &= \mathbb{E}\left[\sum_{t'=1}^{\infty} r_{t'} | s_t = s, \pi\right] \\
&= \mathbb{E}\left[r_1 | \, s_t = s, \pi\right] + \mathbb{E}\left[\sum_{t=2}^{\infty} r_t | s_t = s, \pi\right] \\
&= \mathbb{E}\left[r_1 | \, s_t = s, \pi\right] + \mathbb{E}\left[V^{\pi}(s_{t+1}) | s_t = s, \pi\right]
\end{aligned}
$$

This dynamic consistency is key to many solution approaches.
It states that the value of a state s is related to
the values of its successor states s'.

Markov Decision Problems

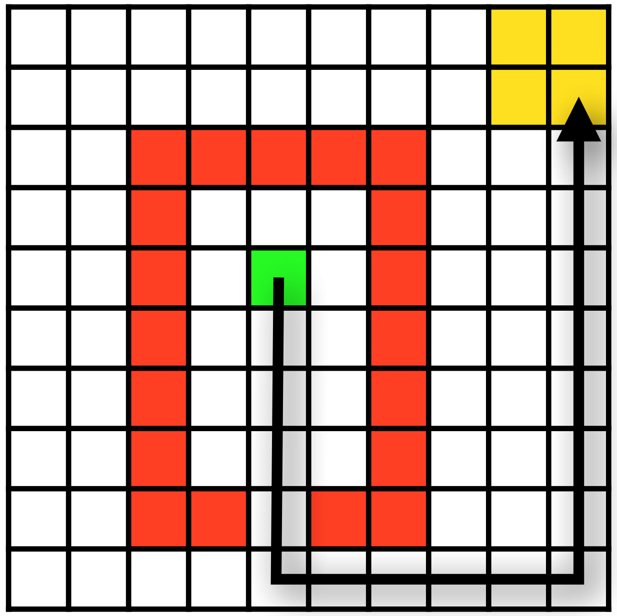


$$V^\pi(s_t) = \boxed{\mathbb{E}\left[r_1 | \, s_t = s, \pi\right]} + \mathbb{E}\left[V(s_{t+1}), \pi\right]$$

$$r_1 \sim \mathcal{R}(s_2, a_1, s_1)$$

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

Bellman equation



$$
\begin{aligned}
V^\pi(s_t) &= \mathbb{E}\left[r_1 | \, s_t = s, \pi\right] + \mathbb{E}\left[V(s_{t+1}), \pi\right] \\
\mathbb{E}\left[r_1 | s_t, \pi\right] &= \sum_a \pi(a, s_t) \left[\sum_{s_{t+1}} \mathcal{T}^a_{s_t s_{t+1}} \mathcal{R}(s_{t+1}, a, s_t)\right] \\
\mathbb{E}\left[V^\pi(s_{t+1}), \pi, s_t\right] &= \sum_a \pi(a, s_t) \left[\sum_{s_{t+1}} \mathcal{T}^a_{s_t s_{t+1}} V^\pi(s_{t+1})\right]
\end{aligned}
$$

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

Bellman Equation

$$\text{All future reward from state s} = E\left[\text{Immediate reward} + \text{All future reward from next state s'}\right]$$

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

Q values = state-action values

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

- so we can define state-action values as:

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

- and state values are average state-action values:

$$V(s) = \sum_a \pi(a|s)\mathcal{Q}(s,a)$$

Bellman Equation

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

- to evaluate a policy, we need to solve the above equation, i.e. find the self-consistent state values

- options for policy evaluation
  - exhaustive tree search - outwards, inwards, depth-first
  - value iteration: iterative updates
  - linear solution in 1 step
  - experience sampling

Solving the Bellman Equation

Option 1: turn it into update equation

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

Option 2: linear solution (w/ absorbing states)

$$\begin{aligned} V(s) &= \sum_a \pi(a, s_t) \left[\sum_{s'} \mathcal{T}^a_{ss'} \left[\mathcal{R}(s', a, s) + V(s') \right] \right] \\ \Rightarrow \mathbf{v} &= \mathbf{R}^\pi + \mathbf{T}^\pi \mathbf{v} \\ \Rightarrow \mathbf{v}^\pi &= (\mathbf{I} - \mathbf{T}^\pi)^{-1} \mathbf{R}^\pi \qquad \mathcal{O}(|\mathcal{S}|^3) \end{aligned}$$

Policy update

Given the value function for a policy, say via linear solution

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

Given the values V for the policy, we can improve the policy by always choosing the best action:

$$\pi'(a|s) = \begin{cases} 1 \,\text{if}\, a = \text{argmax}_a \,\mathcal{Q}^\pi(s,a) \\ 0 \,\text{else} \end{cases}$$

It is guaranteed to improve:

for deterministic policy

$$\mathcal{Q}^\pi(s, \pi'(s)) = \max_a \mathcal{Q}^\pi(s,a) \geq \mathcal{Q}^\pi(s, \pi(s)) = \mathcal{V}^\pi(s)$$

Policy iteration

Policy evaluation

$$\mathbf{v}^\pi \quad = \quad (\mathbf{I} - \mathbf{T}^\pi)^{-1}\mathbf{R}^\pi$$

Value iteration

$$V^*(s) = \max_a \sum_{s'} \mathcal{T}^a_{ss'}\left[\mathcal{R}^a_{ss} + V^*(s')\right]$$

greedy policy improvement

$$\pi(a|s) = \begin{cases} 1 \,\text{if}\, a = \text{argmax}_a \sum_{s'} \mathcal{T}^a_{ss'}\left[\mathcal{R}^a_{ss} + V^{pi}(s')\right] \\ 0 \,\text{else} \end{cases}$$

Model-free solutions

- So far we have assumed knowledge of R and T
  - R and T are the 'model' of the world, so we assume full knowledge of the dynamics and rewards in the environment
- What if we don't know them?
- We can still learn from state-action-reward samples
  - we can learn R and T from them, and use our estimates to solve as above
  - alternatively, we can directly estimate V or Q

Solving the Bellman Equation

Option 3: sampling

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

this is an expectation over policy and transition samples.

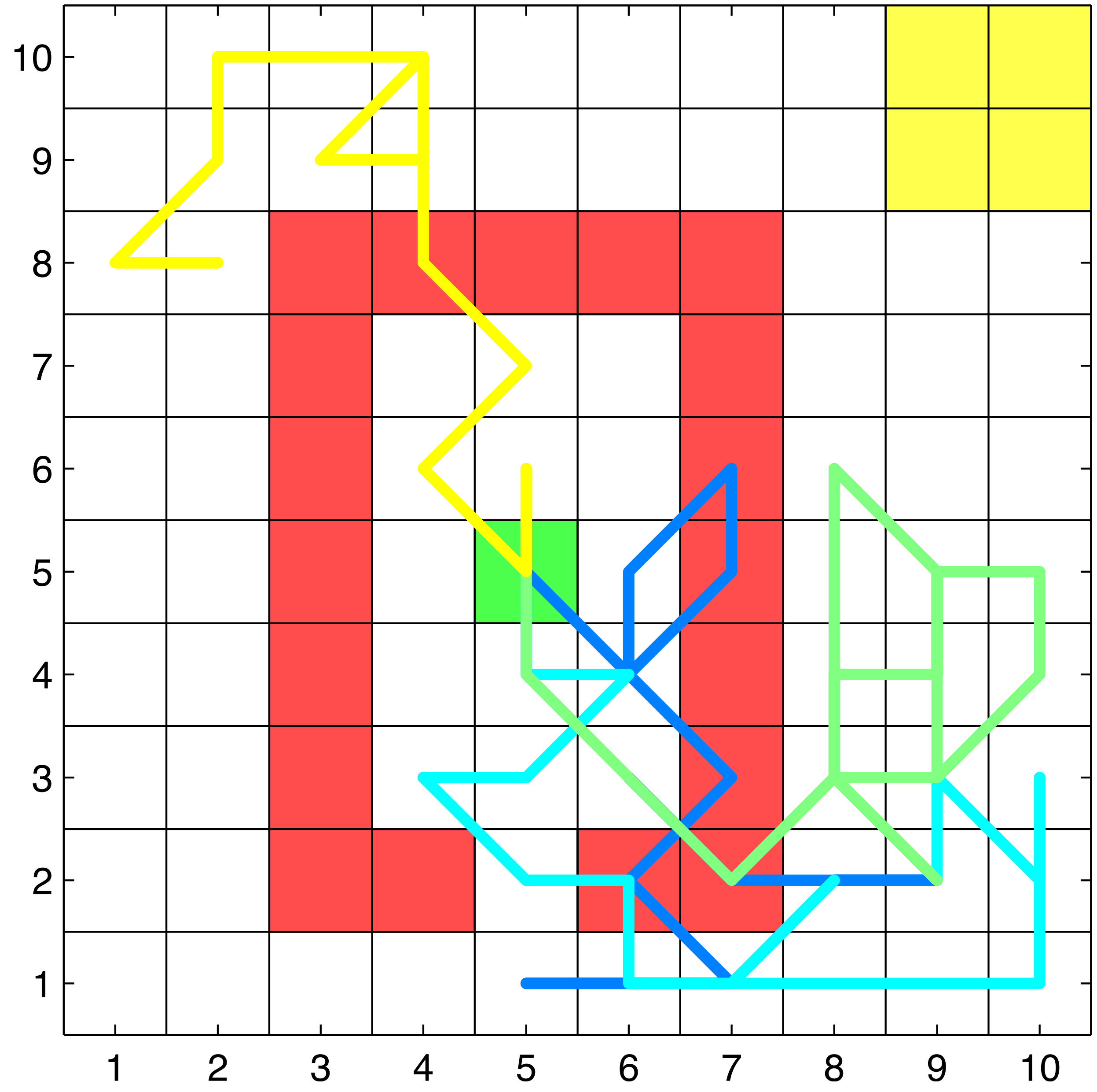
So we can just draw some samples from the policy and the transitions and average over them:

$$a \quad = \quad \sum_k f(x_k) p(x_k)$$

$$x^{(i)} \sim p(x) \rightarrow \hat{a} = \frac{1}{N} \sum_i f(x^{(i)})$$

more about this later...

Learning from samples



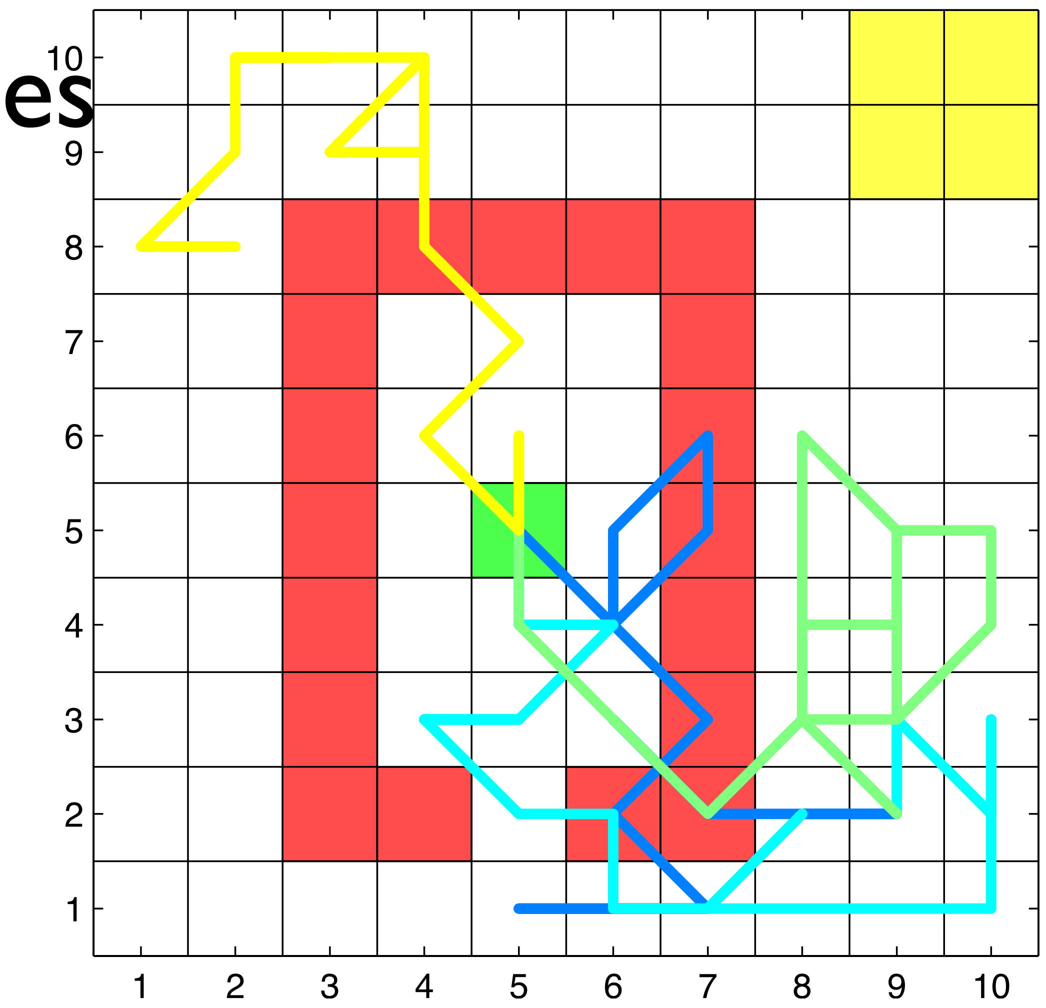
A new problem: exploration versus exploitation

Monte Carlo

- First visit MC
  - randomly start in all states, generate paths, average for starting state only

$$\mathcal{V}(s) = \frac{1}{N} \sum_i \left\{ \sum_{t'=1}^{T} r_{t'}^i | s_0 = s \right\}$$

- More efficient use of samples
  - Every visit MC
  - Bootstrap: TD
  - Dyna
- Better samples
  - on policy versus off policy
  - Stochastic search, UCT...



Update equation: towards TD

Bellman equation

$$V(s) = \sum_a \pi(a,s) \left[\sum_{s'} \mathcal{T}^a_{ss'} \left[\mathcal{R}(s',a,s) + V(s') \right] \right]$$

Not yet converged, so it doesn't hold:

$$dV(s) = -V(s) + \sum_a \pi(a,s) \left[\sum_{s'} \mathcal{T}^a_{ss'} \left[\mathcal{R}(s',a,s) + V(s') \right] \right]$$

And then use this to update

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

TD learning

$$dV(s) = -V(s) + \sum_a \pi(a,s)\left[\sum_{s'} \mathcal{T}^a_{ss'}\left[\mathcal{R}(s',a,s) + V(s')\right]\right]$$

Sample

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

$$\delta_t = -V_{t-1}(s_t) + r_t + V_{t-1}(s_{t+1})$$

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

$$V_t(s_t) = V_{t-1}(s_t) + \alpha\delta_t$$

TD learning

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

$$\delta_t = -V_t(s_t) + r_t + V_t(s_{t+1})$$

$$V_{t+1}(s_t) = V_t(s_t) + \alpha\delta_t$$

SARSA

- Do TD for state-action values instead:

$$\mathcal{Q}(s_t, a_t) \leftarrow \mathcal{Q}(s_t, a_t) + \alpha[r_t + \gamma \mathcal{Q}(s_{t+1}, a_{t+1}) - \mathcal{Q}(s_t, a_t)]$$

$$s_t, a_t, r_t, s_{t+1}, a_{t+1}$$

- convergence guarantees - will estimate $\mathcal{Q}^{\pi}(s, a)$

Q learning: off-policy

- Learn off-policy
  - draw from some policy
  - “only” require extensive sampling

$$\mathcal{Q}(s_t,a_t) \leftarrow \mathcal{Q}(s_t,a_t) + \alpha \left[\underbrace{r_t + \gamma \max_a \mathcal{Q}(s_{t+1},a)}_{\text{update towards optimum}} - \mathcal{Q}(s_t,a_t) \right]$$

- will estimate $\mathcal{Q}^*(s,a)$

The effect of bootstrapping

B1
B1
B1
B1
B1
B1
B0
A0 B0

Markov (every visit)
V(B)=3/4
V(A)=0

TD
V(B)=3/4
V(A)=~3/4

- Average over various bootstrappings: TD(λ)

after Sutton and Barto 1998

Conclusion

- Long-term rewards have internal consistency
- This can be exploited for solution
- Exploration and exploitation trade off when sampling
- Clever use of samples can produce fast learning
  - Brain most likely does something like this

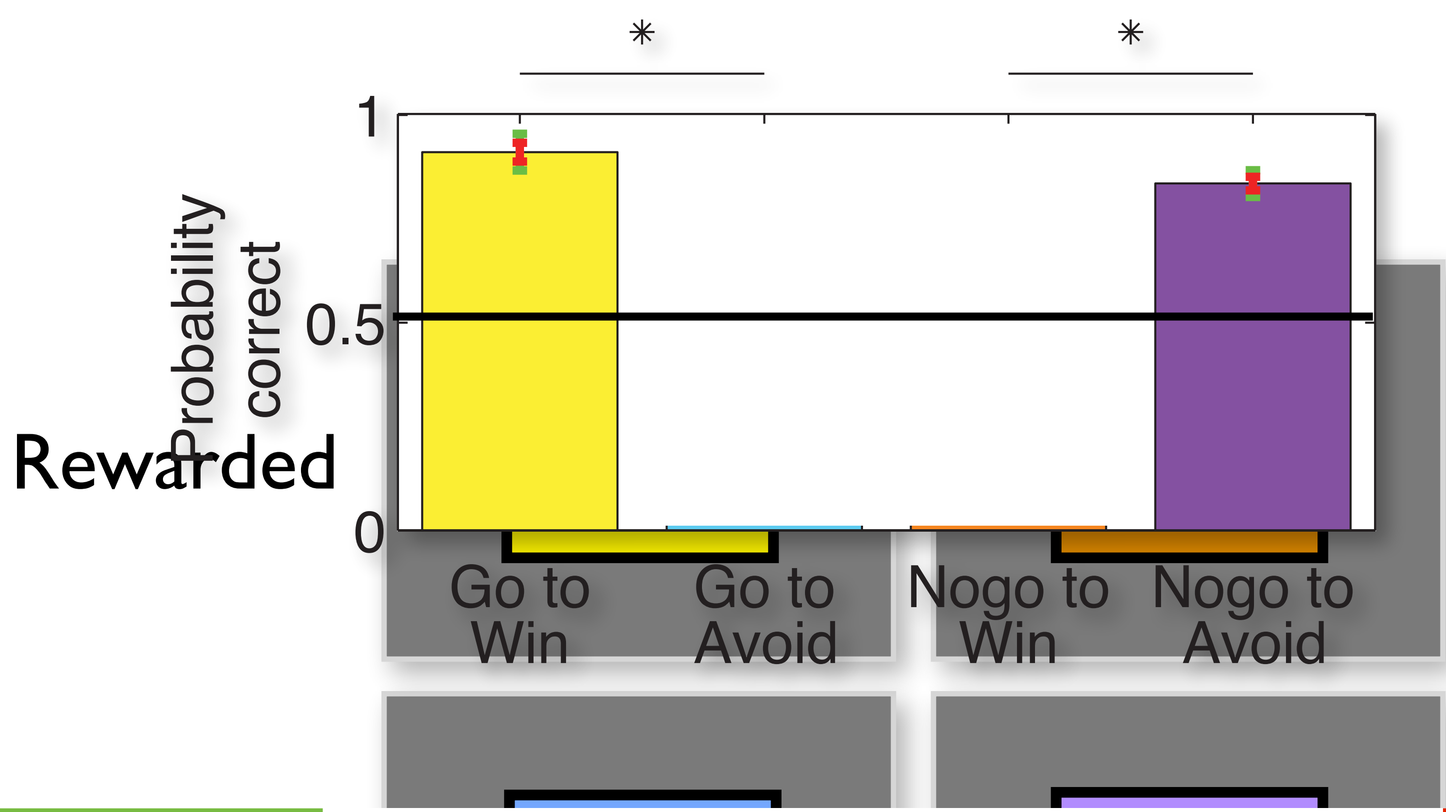
Fitting models to behaviour

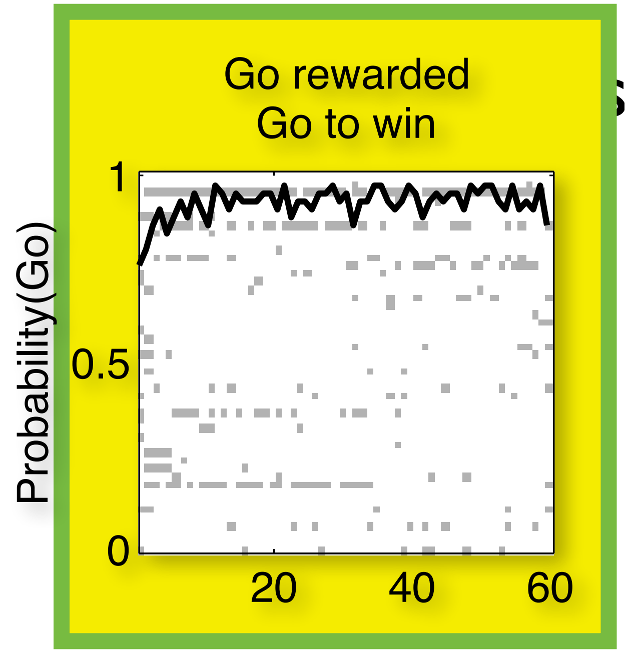
Quentin Huys

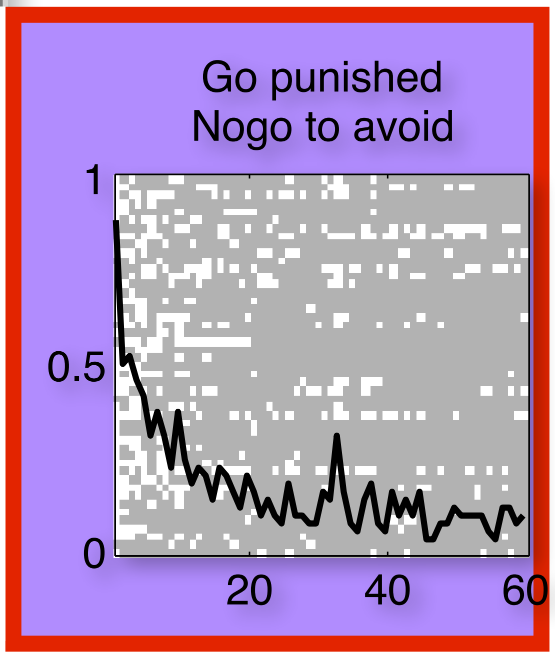
Translational Neuromodeling Unit, University of Zurich and ETH Zurich
University Hospital of Psychiatry Zurich

Computational Psychiatry Course
Zurich, 1.9.2016

Example task





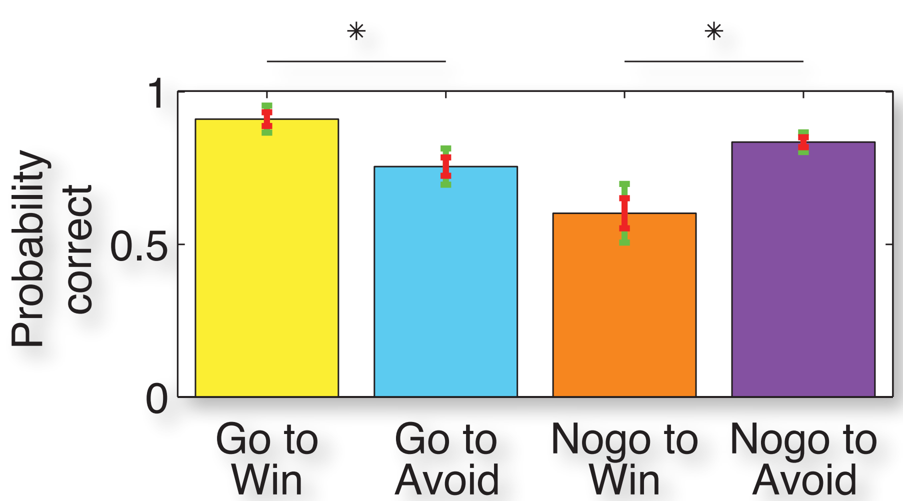


Think of it as four separate two-armed bandit tasks

Guitart-Masip, Huys et al. *2012*

Analysing behaviour

- Standard approach:
  - Decide which feature of the data you care about
  - Run descriptive statistical tests, e.g. ANOVA

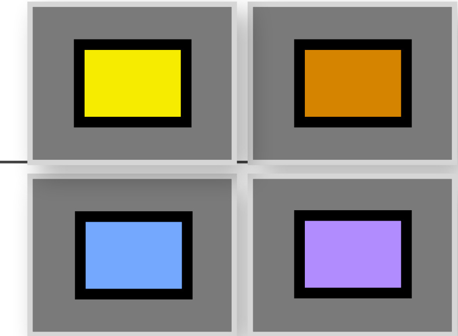


- Many strengths
- Weakness
  - Piecemeal, not holistic / global
  - Descriptive, not generative
  - No internal variables

Models

- Holistic
  - Aim to model the process by which the data came about in its “entirety”
- Generative
  - They can be run on the task to generate data as if a subject had done the task
- Inference process
  - Capture the inference process subjects have to make to perform the task.
  - Do this in sufficient detail to replicate the data.
- Parameters
  - replace test statistics
  - their meaning is explicit in the model

Actions



- Q values "the process"

$$\mathcal{Q}_t(a_t, s_t) = \mathcal{Q}_{t-1}(a_t, s_t) + \epsilon(r_t - \mathcal{Q}_{t-1}(a_t, s_t))$$

- Probabilities "link function"

$$\begin{aligned} p(a_t|s_t, h_t, \beta) &= p(a_t|\mathcal{Q}(a_t, s_t), \beta) \\ &= \frac{e^{\beta\mathcal{Q}(a_t, s_t)}}{\sum_{a'} e^{\beta\mathcal{Q}(a', s_t)}} \end{aligned}$$

- Features:

$$p(a_t|s_t) \propto \mathcal{Q}(a_t, s_t)$$

$$0 \leq p(a) \leq 1$$

- links learning process and observations
  - choices, RTs, or any other data

Fitting models I

- Maximum likelihood (ML) parameters

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta)$$

- where the likelihood of all choices is:

$$\begin{aligned}
\mathcal{L}(\theta) &= \log p(\{a_t\}_{t=1}^T | \{s_t\}_{t=1}^T, \{r_t\}_{t=1}^T, \underbrace{\theta}_{\beta,\epsilon}) \\
&= \log p(\{a_t\}_{t=1}^T | \{\mathcal{Q}(s_t, a_t; \epsilon)\}_{t=1}^T, \beta) \\
&= \log \prod_{t=1}^T p(a_t | \mathcal{Q}(s_t, a_t; \epsilon), \beta) \\
&= \sum_{t=1}^T \log p(a_t | \mathcal{Q}(s_t, a_t; \epsilon), \beta)
\end{aligned}$$

Fitting models II

- No closed form
- Use your favourite method
  - gradients
  - fminunc / fmincon...
- Gradients for RW model

$$
\begin{aligned}
\frac{d\mathcal{L}(\theta)}{d\theta} &= \frac{d}{d\theta}\sum_t \log p(a_t|\mathcal{Q}_t(a_t,s_t;\epsilon),\beta) \\
&= \sum_t \frac{d}{d\theta}\beta\mathcal{Q}_t(a_t,s_t;\epsilon) - \sum_{a'} p(a'|\mathcal{Q}_t(a',s_t;\epsilon),\beta)\frac{d}{d\theta}\beta\mathcal{Q}_t(a',s_t;\epsilon) \\
\frac{d\mathcal{Q}_t(a_t,s_t;\epsilon)}{d\epsilon} &= (1-\epsilon)\frac{d\mathcal{Q}_{t-1}(a_t,s_t;\epsilon)}{d\epsilon} + (r_t - \mathcal{Q}_{t-1}(a_t,s_t;\epsilon))
\end{aligned}
$$

Little tricks

- Transform your variables

$$
\begin{aligned}
\beta &= e^{\beta'} \\
&\Rightarrow \quad \beta' = \log(\beta) \\
\epsilon &= \frac{1}{1+e^{-\epsilon'}} \\
&\Rightarrow \quad \epsilon' = \log\left(\frac{\epsilon}{1-\epsilon}\right)
\end{aligned}
$$

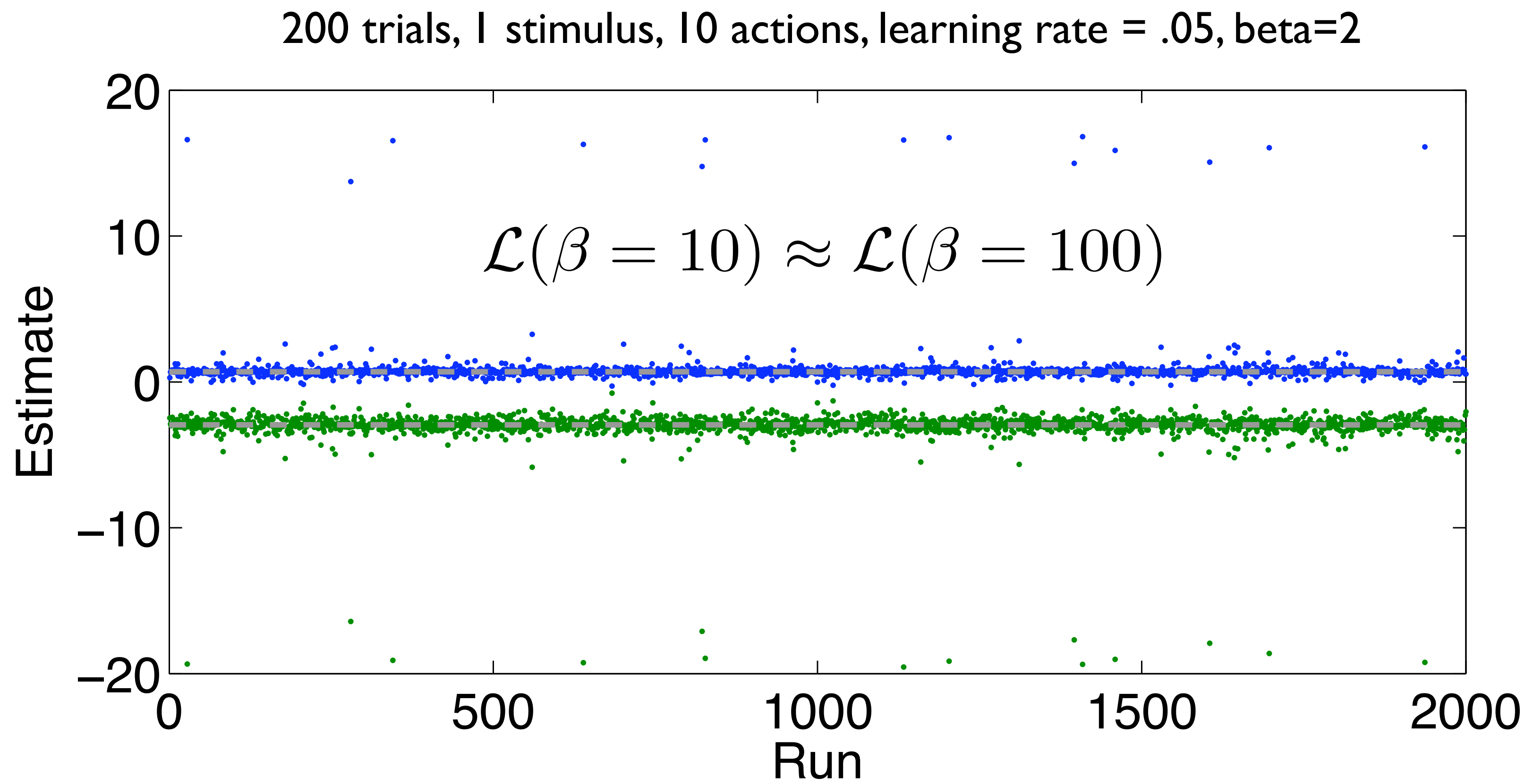
$$\frac{d \log \mathcal{L}(\theta')}{d\theta'}$$

- Avoid over/underflow

$$
\begin{aligned}
y(a) &= \beta Q(a) \\
y_m &= \max_a y(a) \\
p &= \frac{e^{y(a)}}{\sum_b e^{y(b)}} = \frac{e^{y(a)-y_m}}{\sum_b e^{y(b)-y_m}}
\end{aligned}
$$

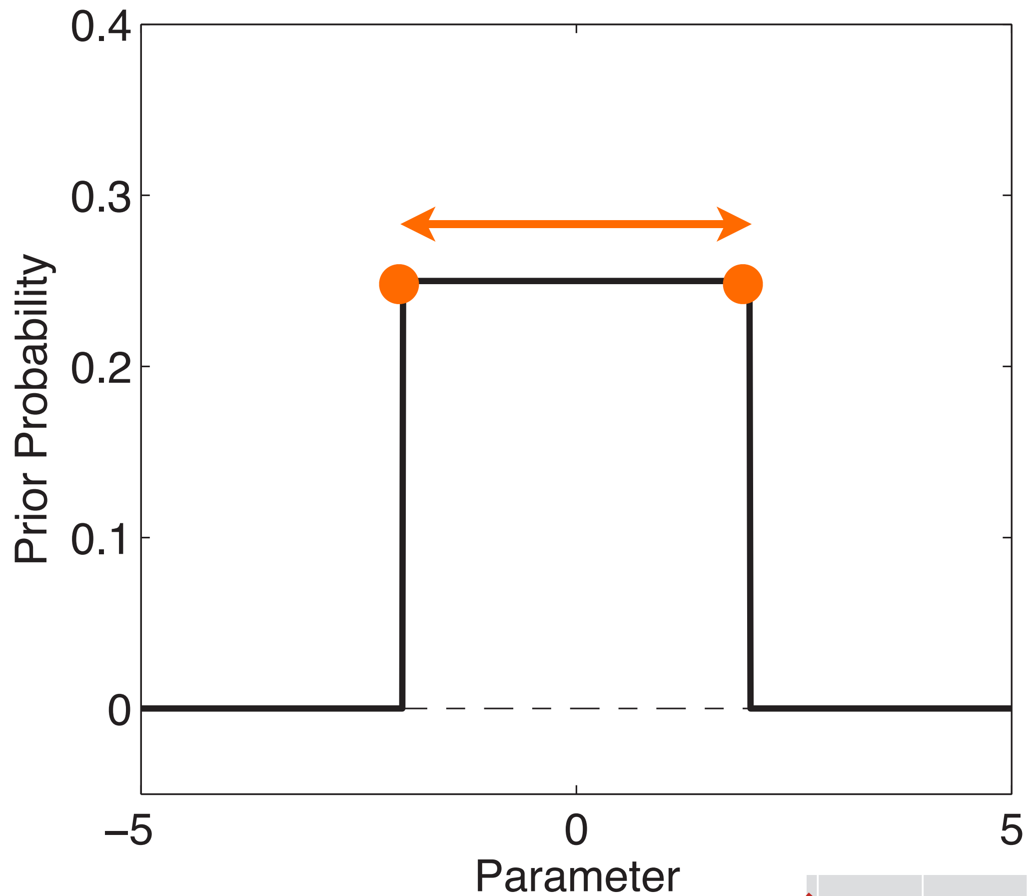
ML characteristics

- ML is asymptotically consistent, but variance high
  - 10-armed bandit, infer beta and epsilon

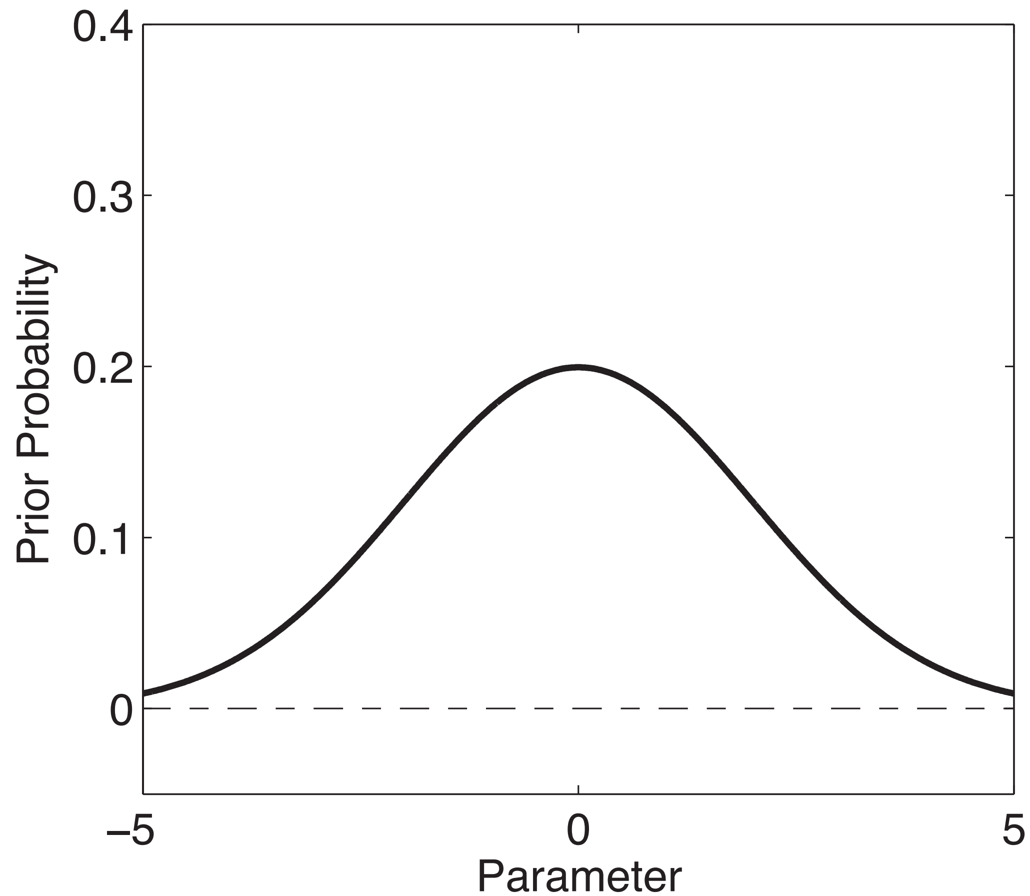


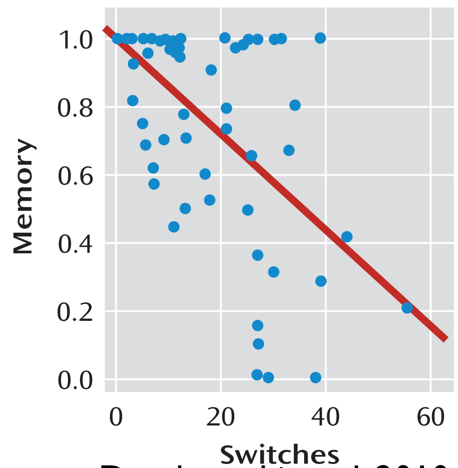
Priors

Not so smooth



Smooth





Dombrovski et al. 2010

Maximum a posteriori estimate

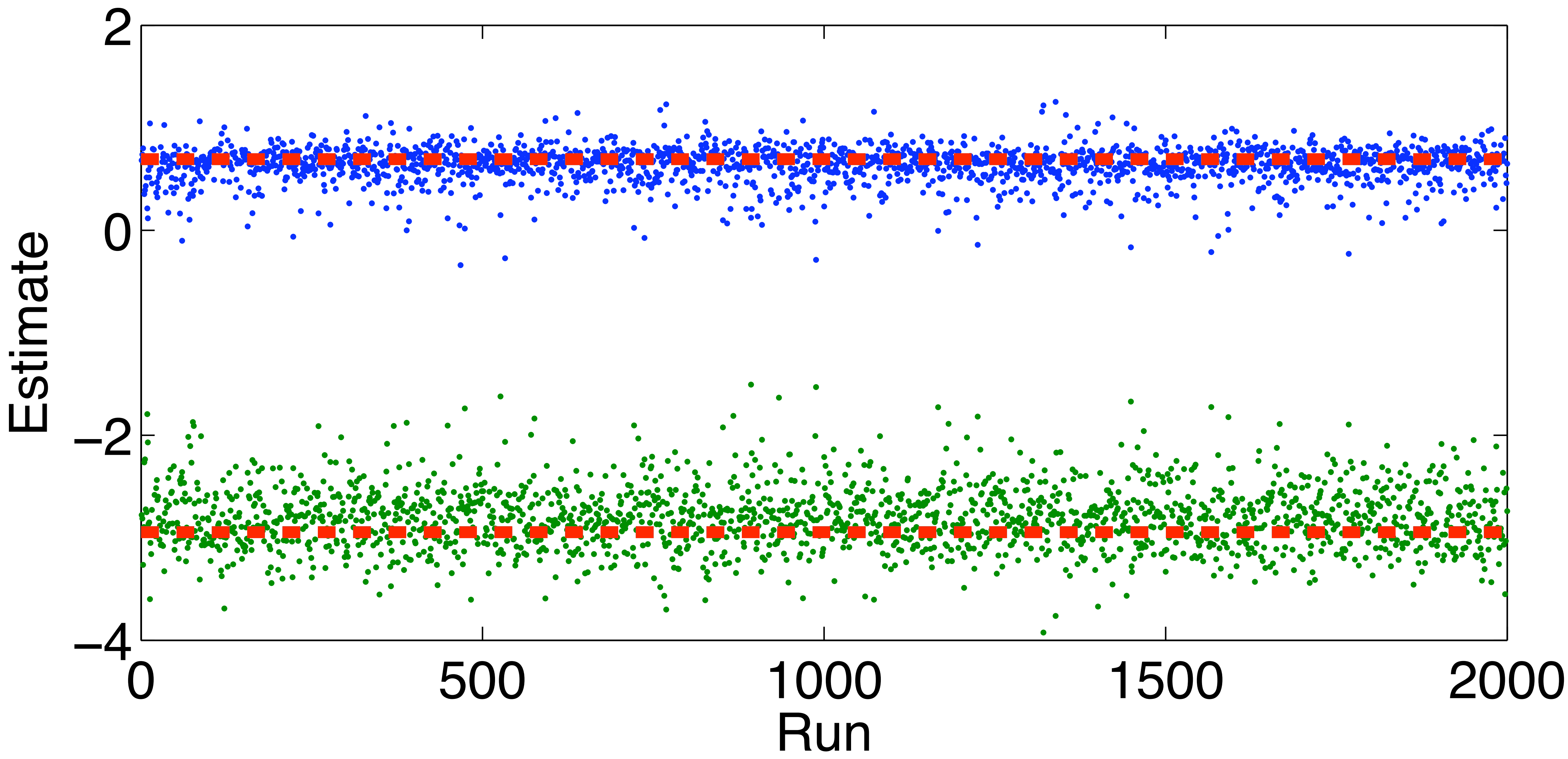
$$\mathcal{P}(\theta) = p(\theta|a_{1\ldots T}) = \frac{p(a_{1\ldots T}|\theta)p(\theta)}{\int d\theta p(\theta|a_{1\ldots T})p(\theta)}$$

$$\log \mathcal{P}(\theta) = \sum_{t=1}^{T} \log p(a_t|\theta) + \log p(\theta) + const.$$

$$\frac{\log \mathcal{P}(\theta)}{d\alpha} = \frac{\log \mathcal{L}(\theta)}{d\alpha} + \frac{d\, p(\theta)}{d\theta}$$

- If likelihood is strong, prior will have little effect
  - mainly has influence on poorly constrained parameters
  - if a parameter is strongly constrained to be outside the typical range of the prior, then it will win over the prior

Maximum a posteriori estimate



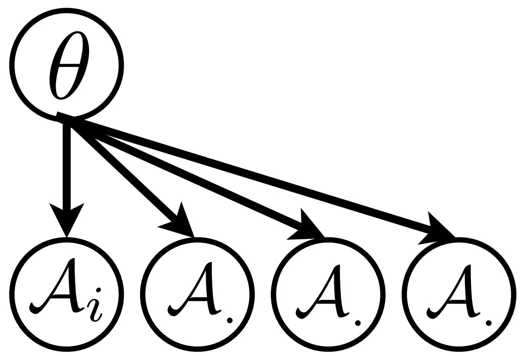
200 trials, 1 stimulus, 10 actions, learning rate = .05, beta=2

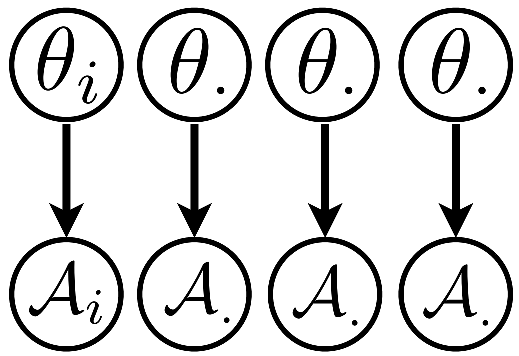
m_{beta}=0, m_{eps}=-3, n=1

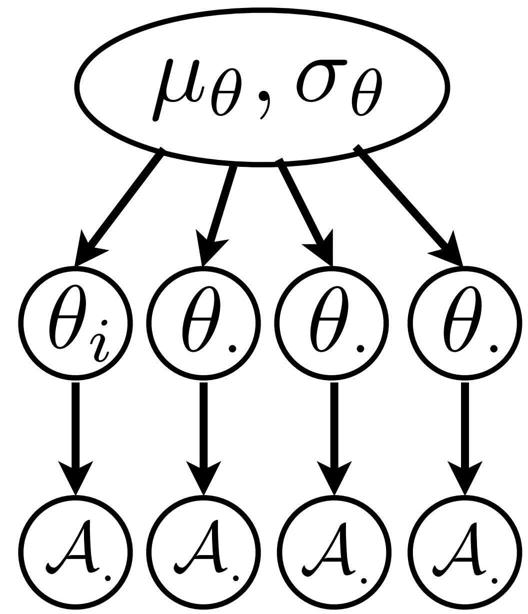
But

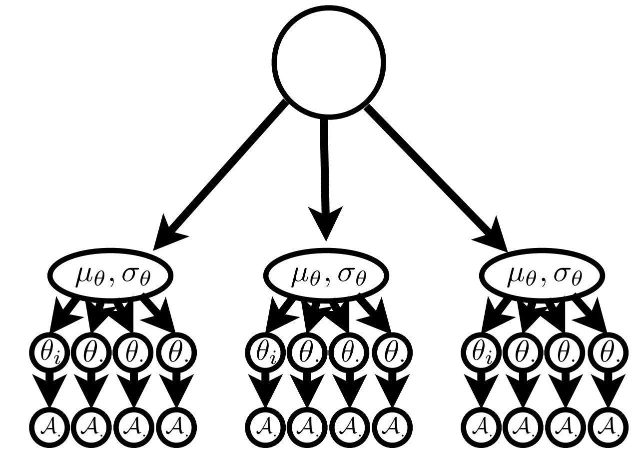
What prior parameters should I use?

Hierarchical estimation - “random” effects









- Fixed effect
  - conflates within- and between- subject variability
- Average behaviour
  - disregards between-subject variability
  - need to adapt model
- Summary statistic
  - treat parameters as random variable, one for each subject
  - overestimates group variance as ML estimates noisy
- Random effects
  - prior mean = group mean

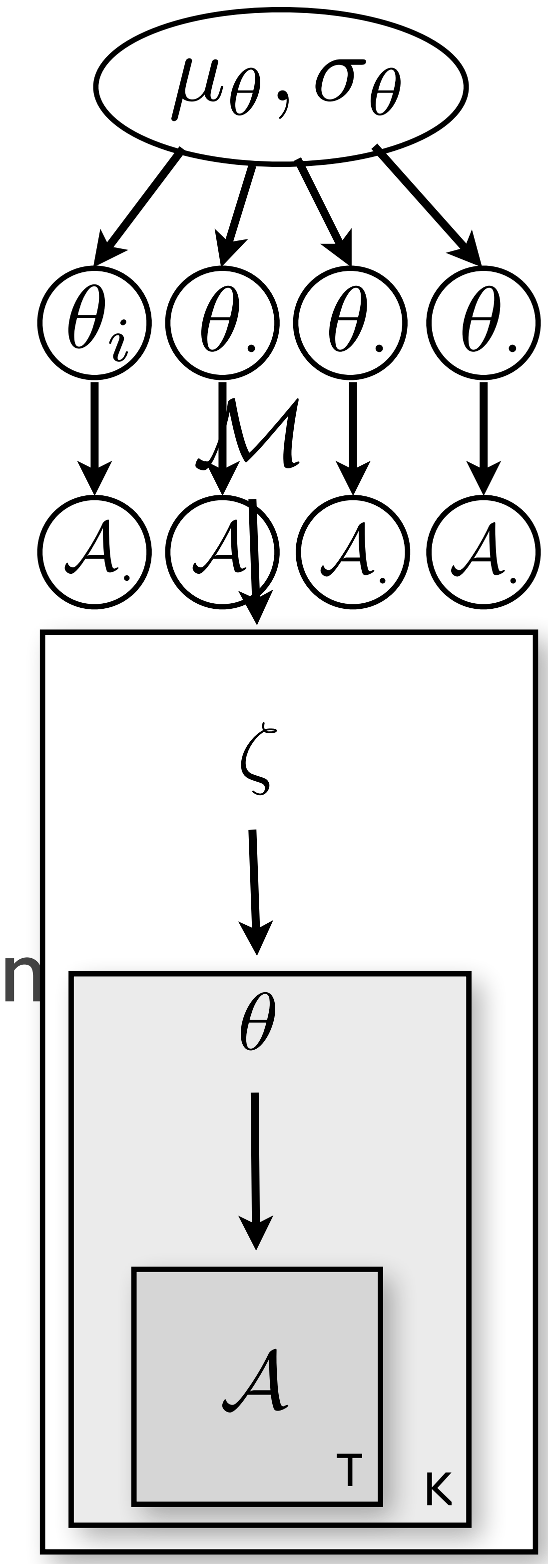
$$p(\mathcal{A}_i|\mu_\theta,\sigma_\theta) = \int d\theta_i\, p(\mathcal{A}_i|\theta_i)\, p(\theta_i|\underbrace{\mu_\theta,\sigma_\theta}_{\zeta})$$

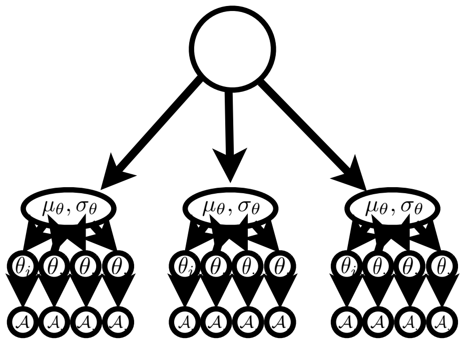
Random effects

- See subjects as drawn from group
- Fixed models
  - all the same: fixed effect wrt model
  - parametrically nested

$$Q(a,s) = \omega_1 Q^1(a,s) + \omega_2 Q^2(a,s)$$

  - assumes within-subject mixture, rather than mixture of perfect types
- Random effects in models





Estimating the hyperparameters

- Effectively we now want to do gradient ascent on:

$$\frac{d}{d\zeta}p(\mathcal{A}|\zeta)$$

- But this contains an integral over individual parameters:

$$p(\mathcal{A}|\zeta) = \int d\theta p(\mathcal{A}|\theta)\, p(\theta|\zeta)$$

- So we need to:

$$\begin{aligned}\hat{\zeta} &= \operatorname*{argmax}_{\zeta} p(\mathcal{A}|\zeta) \\ &= \operatorname*{argmax}_{\zeta} \int d\,\theta p(\mathcal{A}|\theta)\, p(\theta|\zeta)\end{aligned}$$

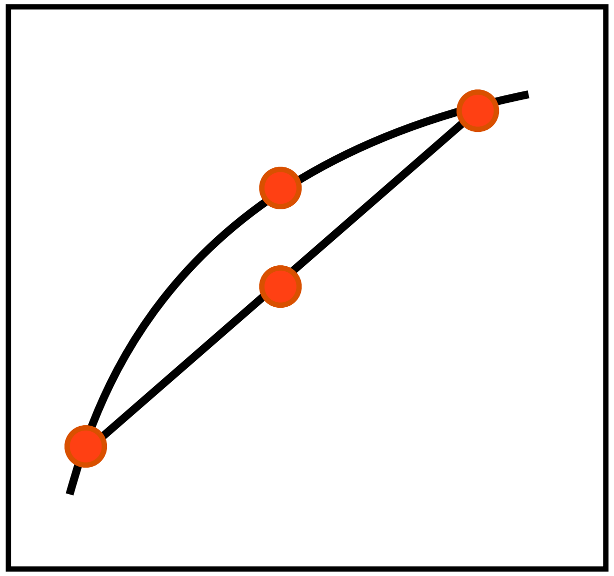
Inference

$$
\begin{aligned}
\hat{\zeta} &= \underset{\zeta}{\operatorname{argmax}}\, p(\mathcal{A}|\zeta) \\
&= \underset{\zeta}{\operatorname{argmax}} \int d\,\theta p(\mathcal{A}|\theta)\, p(\theta|\zeta)
\end{aligned}
$$

- analytical - rare
- brute force - for simple problems
- Expectation Maximisation - approximate, easy
- Variational Bayes
- Sampling / MCMC

Expectation Maximisation

$$
\begin{aligned}
\log p(\mathcal{A}|\zeta) &= \log \int d\theta\, p(\mathcal{A},\theta|\zeta) \\
&= \log \int d\theta\, q(\theta) \frac{p(\mathcal{A},\theta|\zeta)}{q(\theta)} \\
&\geq \int d\theta\, q(\theta) \log \frac{p(\mathcal{A},\theta|\zeta)}{q(\theta)} \\
k^{\text{th}} \text{ E step: } q^{(k+1)}(\theta) &\leftarrow p(\theta|\mathcal{A},\zeta^{(k)}) \\
k^{\text{th}} \text{ M step: } \zeta^{(k+1)} &\leftarrow \underset{\zeta}{\operatorname{argmax}} \int d\theta\, q(\theta) \log p(\mathcal{A},\theta|\zeta)
\end{aligned}
$$

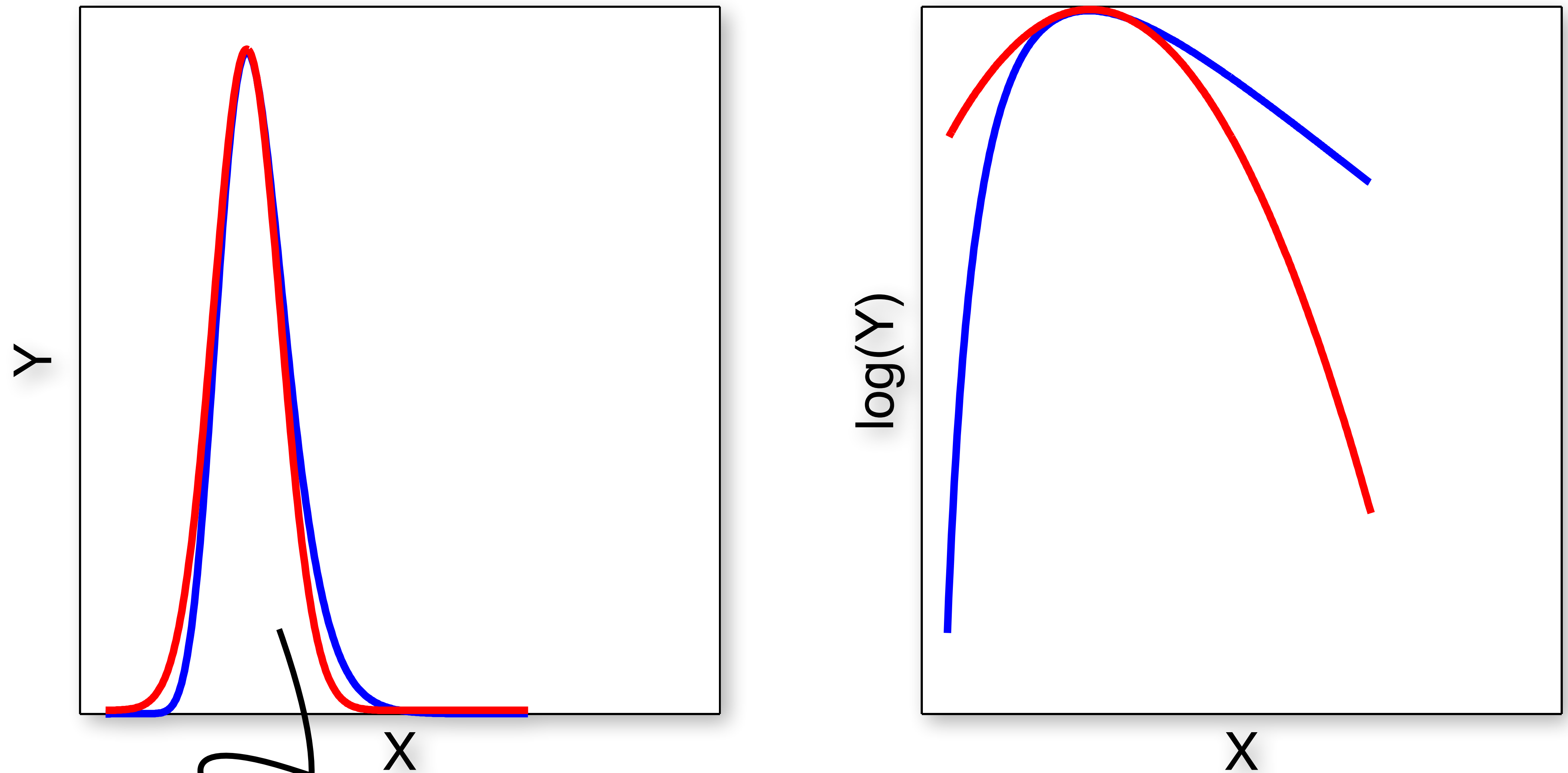


Jensen's inequality

- Iterate between
  - Estimating MAP parameters given prior parameters
  - Estimating prior parameters from MAP parameters

Bayesian Information Criterion

- Laplace's approximation (saddle-point method)



Just a Gaussian

$$\int dx\, f(x) \approx f^*(x_0)\,\sqrt{2\pi\sigma^2}$$

EM with Laplace approximation

- **E step:** $q^{(k+1)}(\theta) \leftarrow p(\theta|\mathcal{A}, \zeta^{(k)})$
  - only need sufficient statistics to perform M step
  - Approximate $p(\theta|\mathcal{A}, \zeta^{(k)}) \sim \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k)$
  - and hence:

$$\begin{aligned}
\text{E step:} \quad q_k(\theta) &= \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k) \\
\mathbf{m}_k &\leftarrow \operatorname*{argmax}_\theta p(\mathbf{a}_k|\theta) p(\theta|\zeta^{(i)}) \\
\mathbf{S}_k^{-1} &\leftarrow \left. \frac{\partial^2 p(\mathbf{a}^k|\theta) p(\theta|\zeta^{(i)})}{\partial \theta^2} \right|_{\theta = \mathbf{m}_k}
\end{aligned}$$

matlab: `[m,L,,,S]=fminunc(…)`

Just what we had before: MAP inference given some prior parameters

EM with Laplace approximation

- Updates

Prior mean = mean of MAP estimates

$$\text{M step:} \qquad \zeta_{\mu}^{(i+1)} = \frac{1}{K}\sum_k \mathbf{m}_k$$

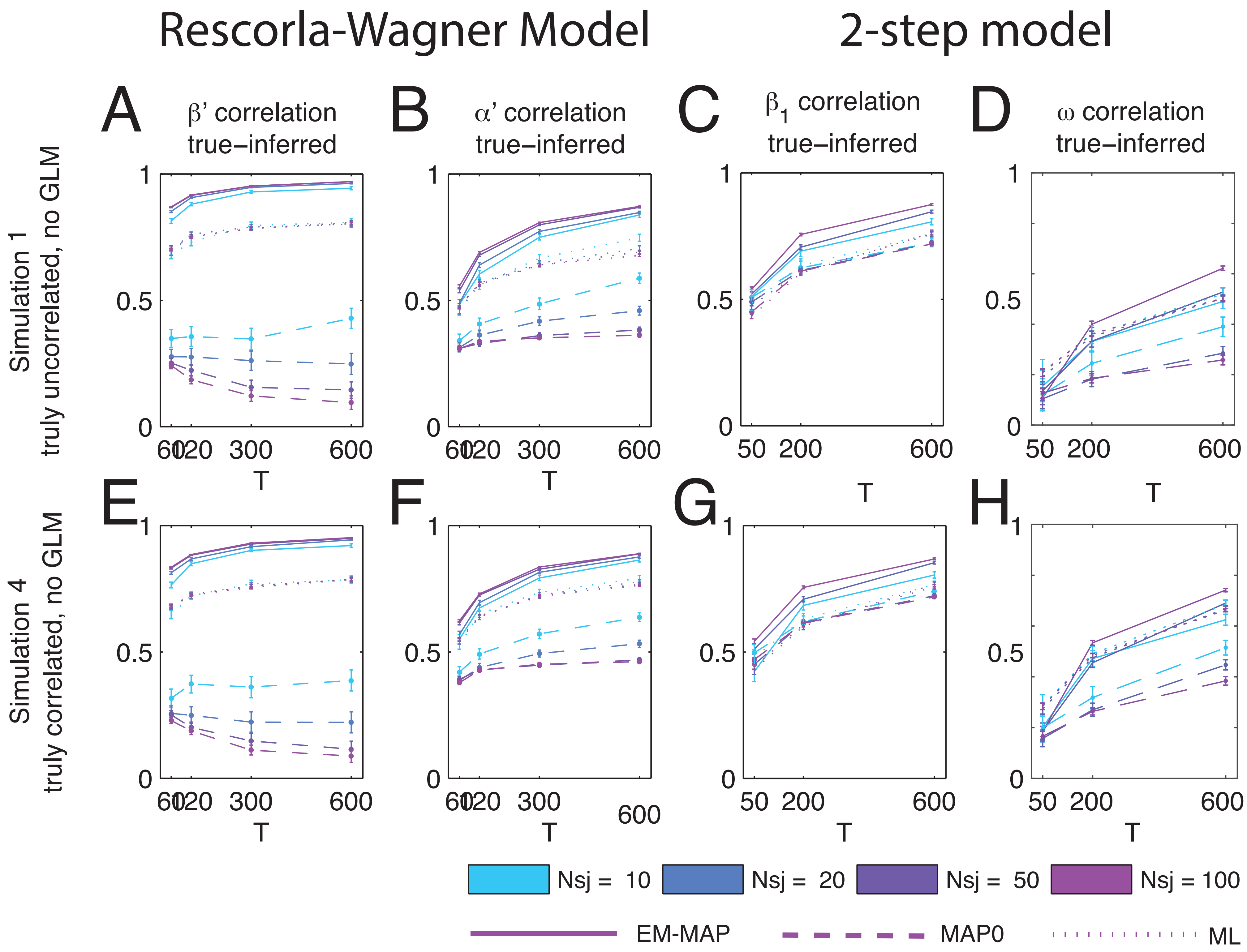
$$\zeta_{\nu^2}^{(i+1)} = \frac{1}{N}\sum_i \left[(\mathbf{m}_k)^2 + \mathbf{S}_k\right] - (\zeta_{\mu}^{(i+1)})^2$$

Prior variance depends on inverse Hessian S and variance of MAP estimates

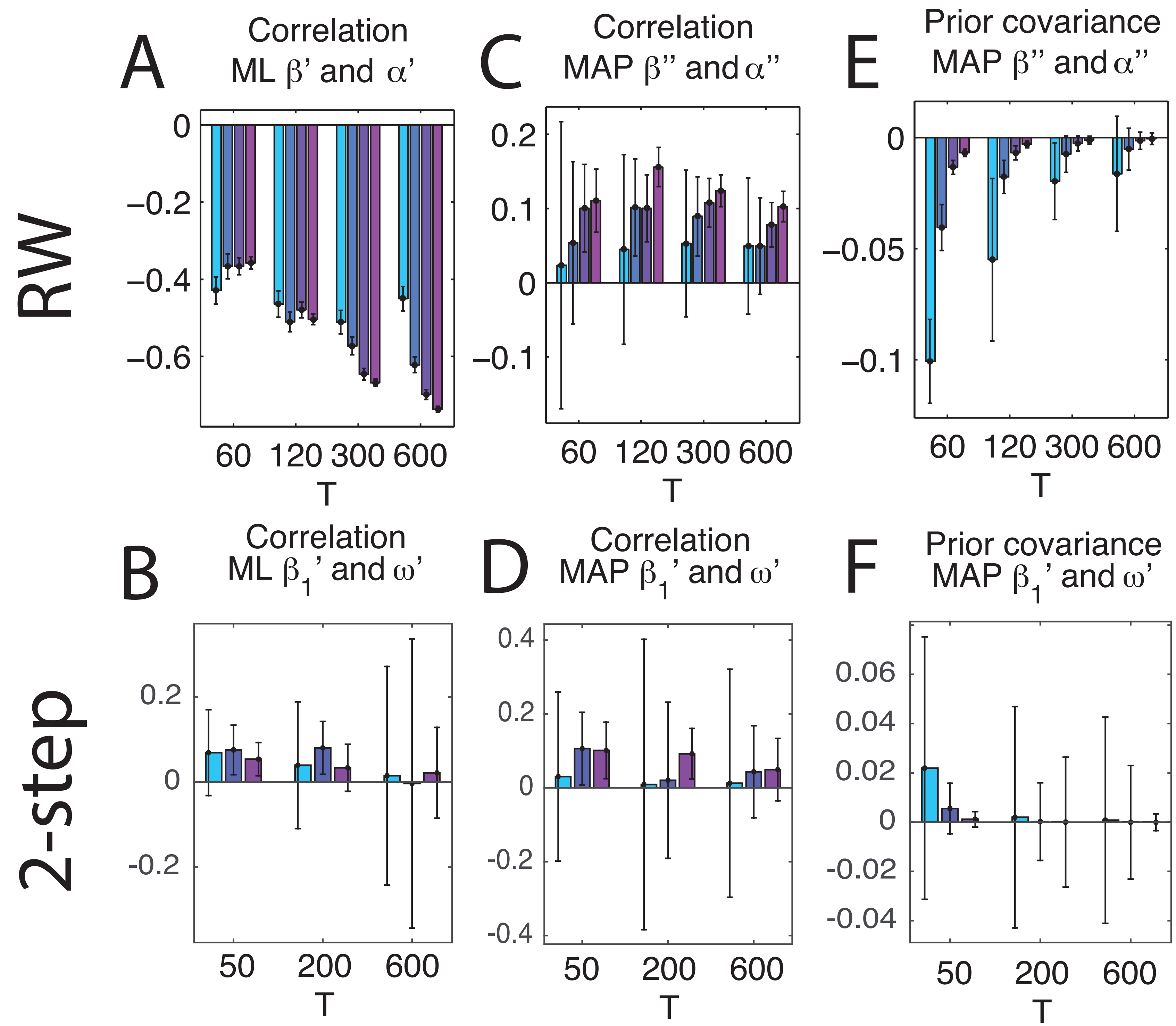
Take uncertainty of estimates into account

- And now iterate until convergence

Parameter recovery

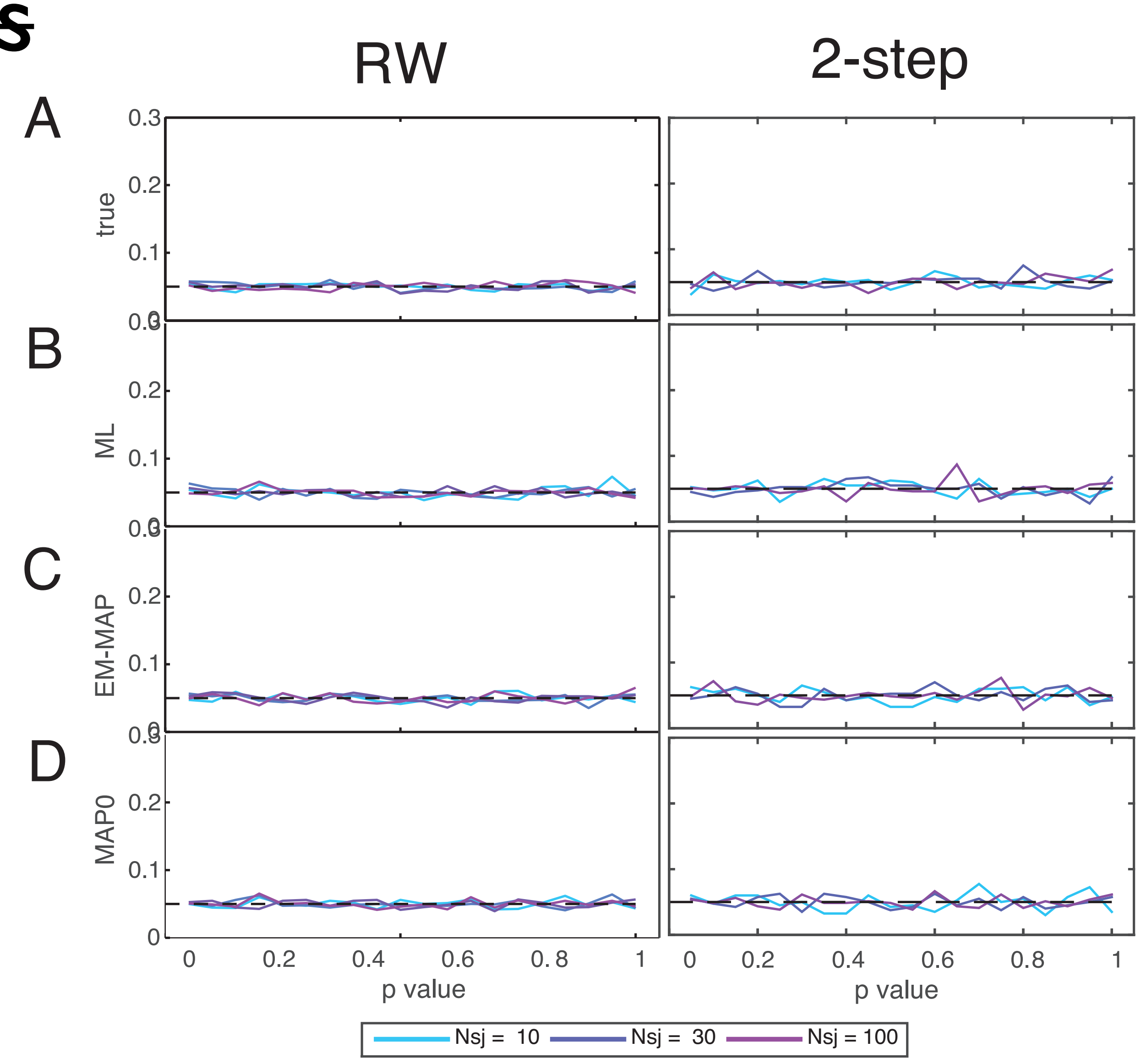


Correlations



Are parameters ok for correlations?

- Individual subject parameter estimates NO LONGER INDEPENDENT!
  - Change group -> change parameter estimates
- ~~compare different params~~
  - if different priors
- correlations, t-tests
  - within same prior ok

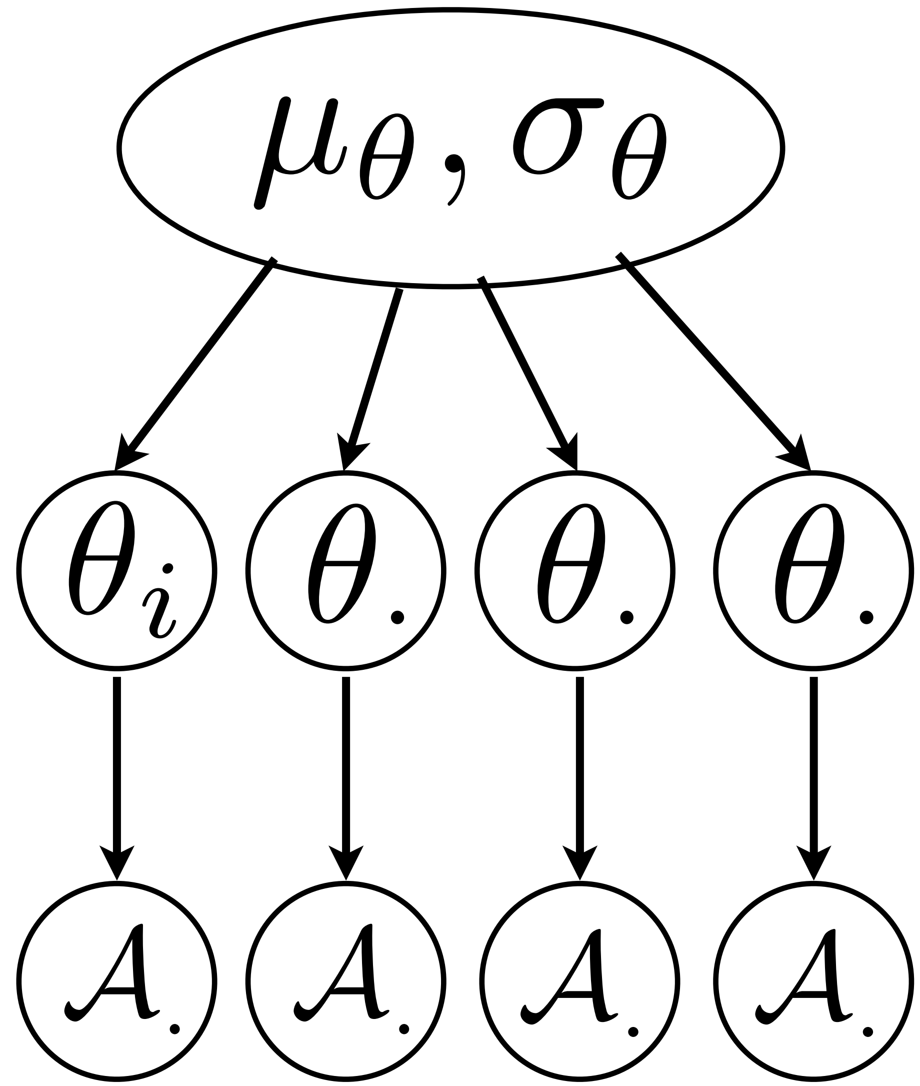


GLM

- So far
  - infer individual parameters
  - apply standard tests
- Alternative
  - View as variation across group
  - Specific - more powerful?

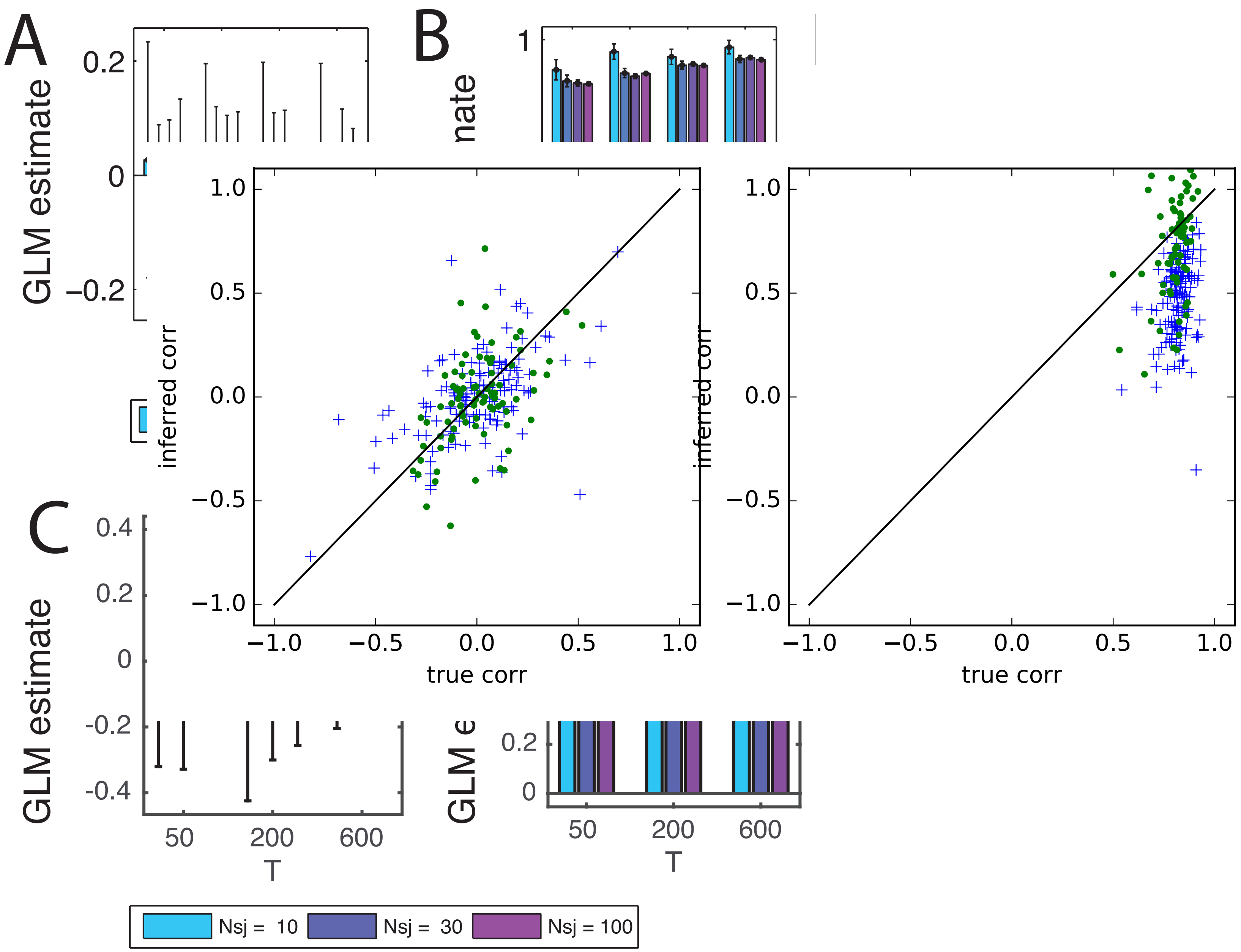
$$\mu_\theta^i = \mu_\theta^{\text{Group}} + \beta\psi_i$$

Infer



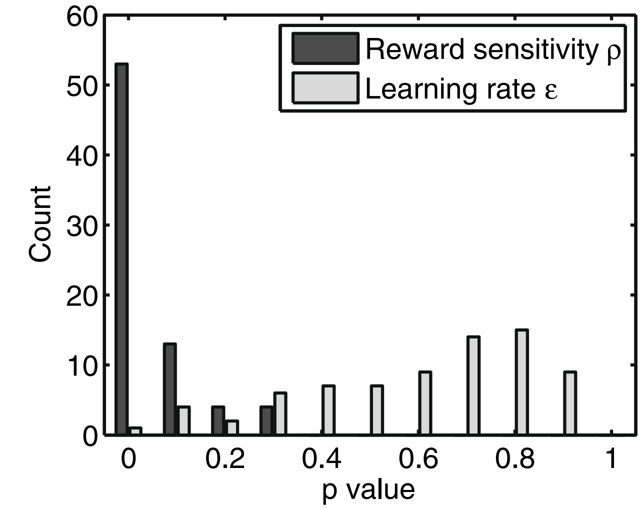
GLM

- Group-level regressor



Fitting - how to

- Write your likelihood function
  - matlab examples attached with emfit.m
    - don't do 20 ML fits!
  - pass it into emfit.m or julia version
    - www.quentinhuys.com/pub/emfit_151110.zip
  - validate: generate data with fitted params
    - compare, have a look, does it look right?
    - re-fit - is it stable?
  - model comparison
  - now: look at parameters, do correlations etc.

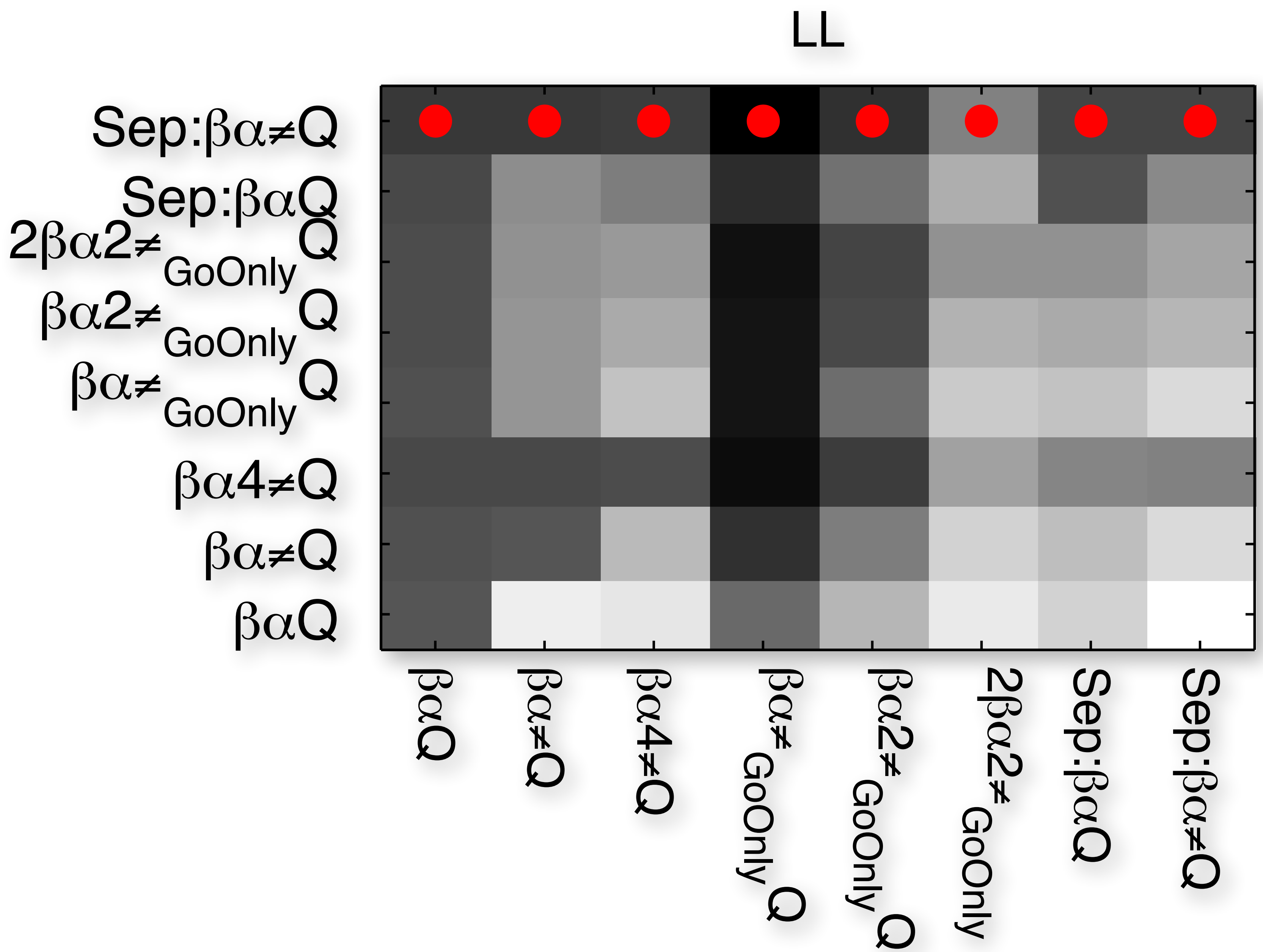


- Future:
  - GLM
  - full random effects over models and parameters jointly?
    - Daniel Schad

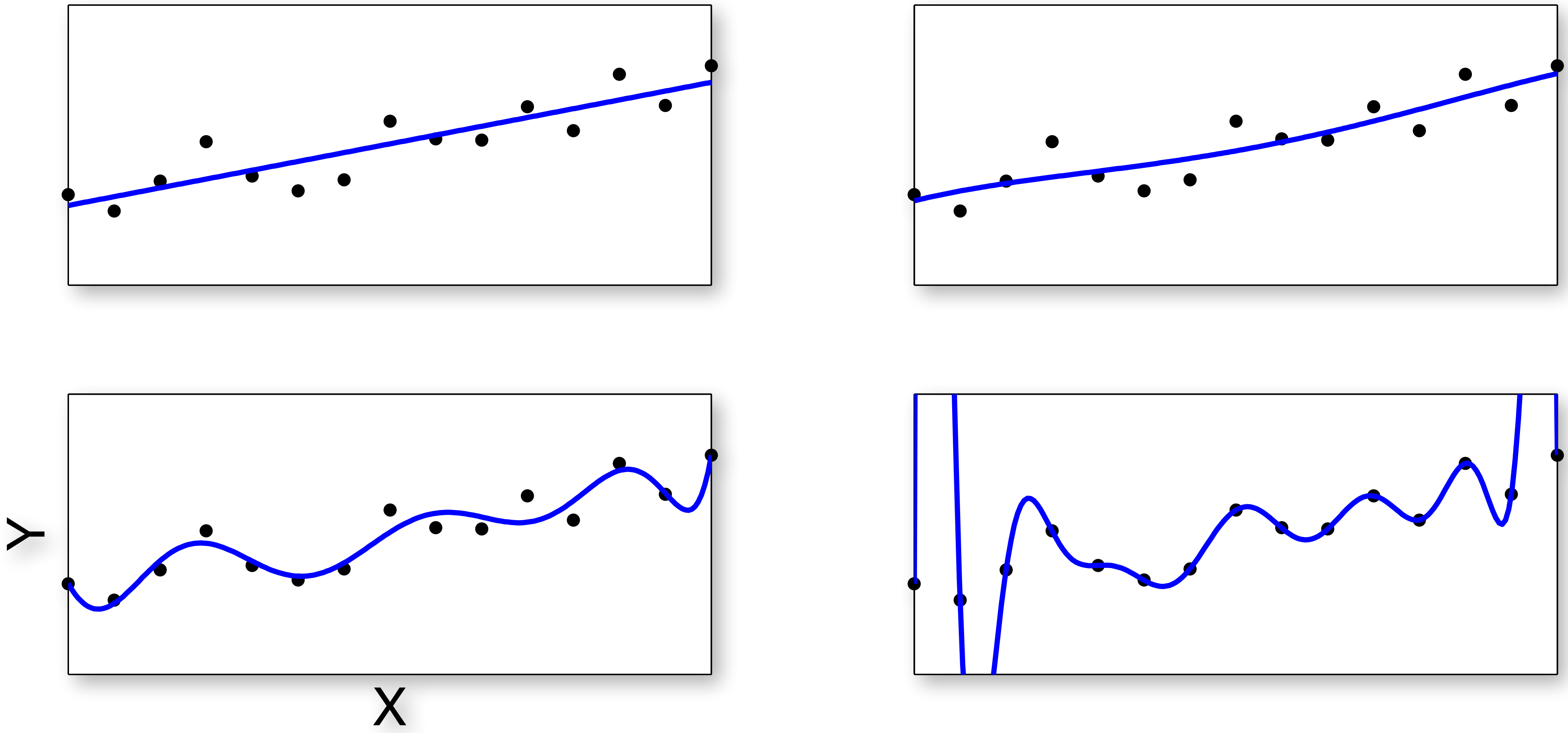
Hierarchical / random effects models

- Advantages
  - Accurate group-level mean and variance
  - Outliers due to weak likelihood are regularised
  - Strong outliers are not
  - Useful for model selection
- Disadvantages
  - Individual estimates θ_i depend on other data, i.e. on $\mathcal{A}_{j\neq i}$ and therefore need to be careful in interpreting these as summary statistics
  - More involved; less transparent
- Psychiatry
  - Groups often not well defined, covariates better
- fMRI
  - Shrink variance of ML estimates - fixed effects better still?

How does it do?



Overfitting



Model comparison

- A fit by itself is not meaningful
- Generative test
  - qualitative
- Comparisons
  - vs random
  - vs other model -> test specific hypotheses and isolate particular effects in a generative setting

Model comparison

- Averaged over its parameter settings, how well does the model fit the data?

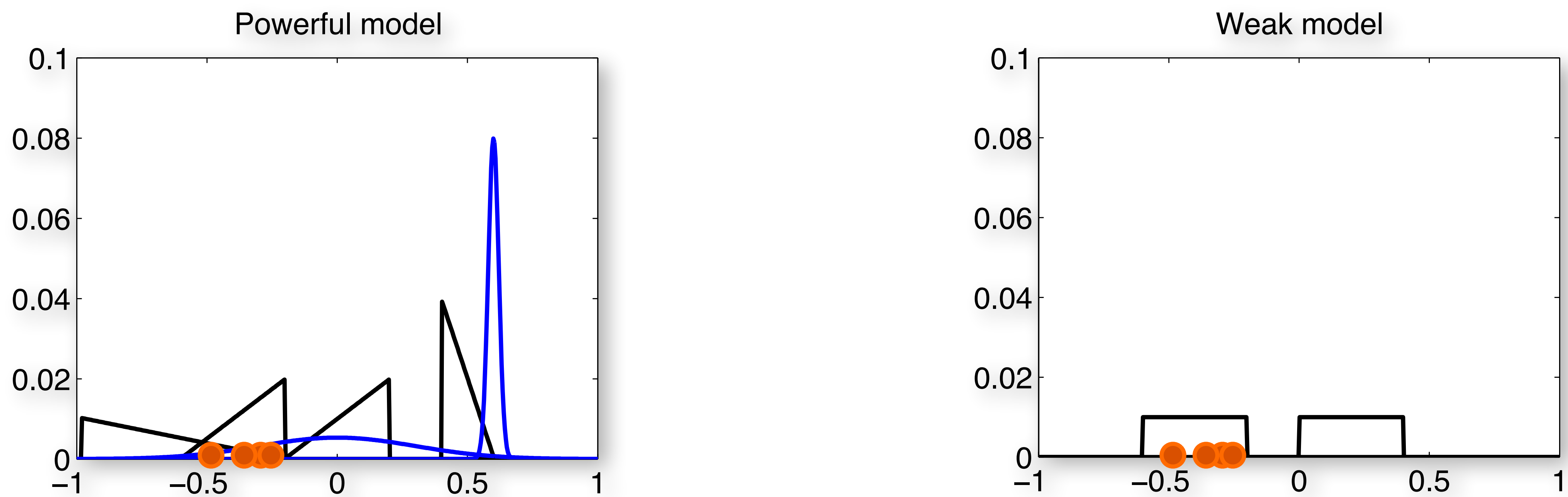
$$p(\mathcal{A}|\mathcal{M}) = \int d\theta \, p(\mathcal{A}|\theta) \, p(\theta|\mathcal{M})$$

- Model comparison: Bayes factors

$$BF = \frac{p(\mathcal{A}|\mathcal{M}_1)}{p(\mathcal{A}|\mathcal{M}_2)}$$

- Problem:
  - integral rarely solvable
  - approximation: Laplace, sampling, variational...

Why integrals? The God Almighty test



$$\frac{1}{N}\left(\mathbf{p}(\mathbf{X}|\boldsymbol{\theta}_\mathbf{1}) + p(X|\theta_2) + \cdots\right)$$

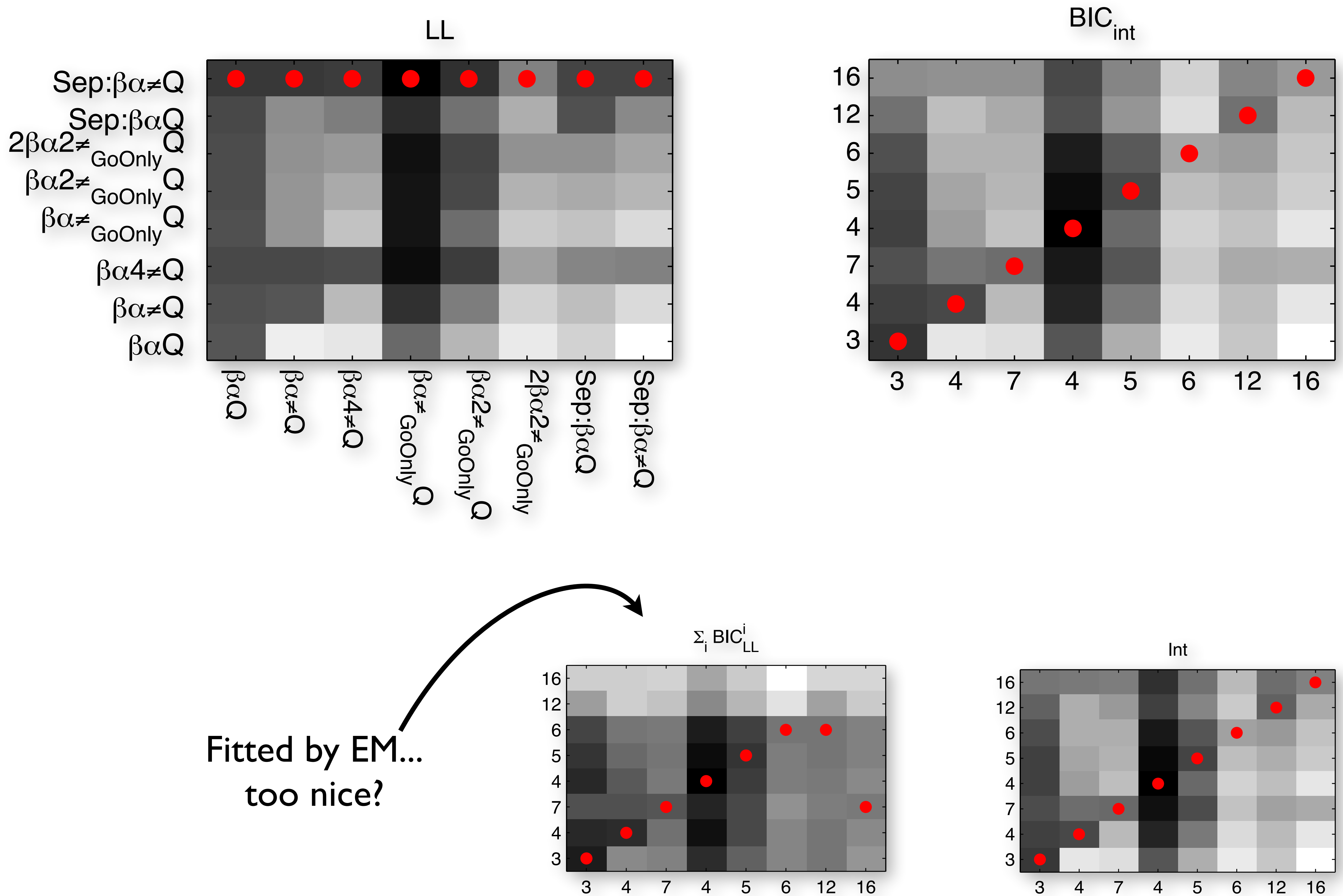
These two factors fight it out
Model complexity vs model fit

Group-level BIC

$$
\begin{aligned}
\log p(\mathcal{A}|\mathcal{M}) &= \int d\zeta\, p(\mathcal{A}|\zeta)\, p(\zeta|\mathcal{M}) \\
&\approx -\frac{1}{2}\mathsf{BIC}_{\mathsf{int}} \\
&= \log \hat{p}(\mathcal{A}|\hat{\zeta}^{ML}) - \frac{1}{2}|\mathcal{M}|\log(|\mathcal{A}|)
\end{aligned}
$$

- Very simple
  - 1) EM to estimate group prior mean & variance
    - simply done using fminunc, which provides Hessians
  - 2) Sample from estimated priors
  - 3) Average

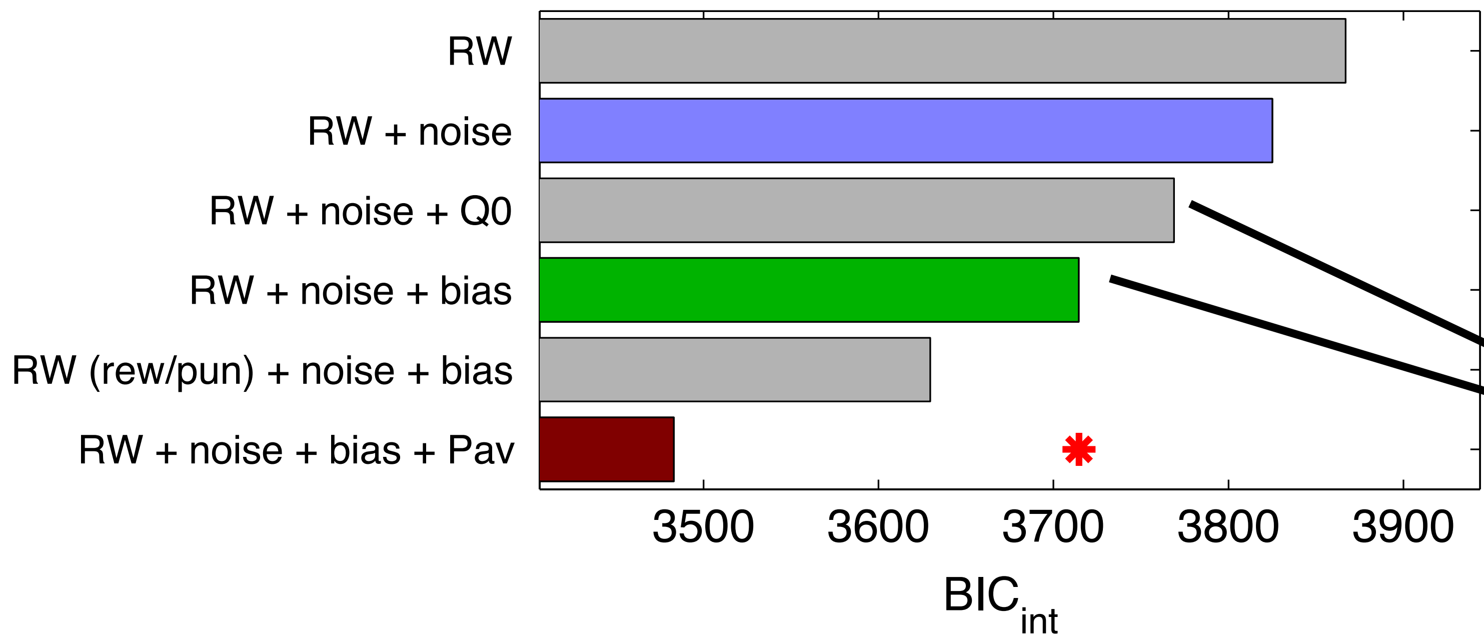
How does it do?

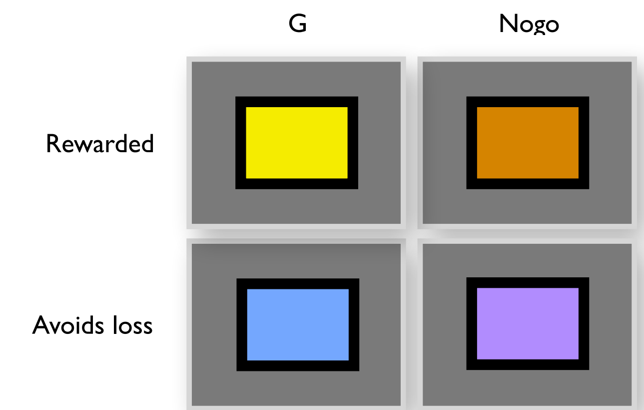


Group Model selection

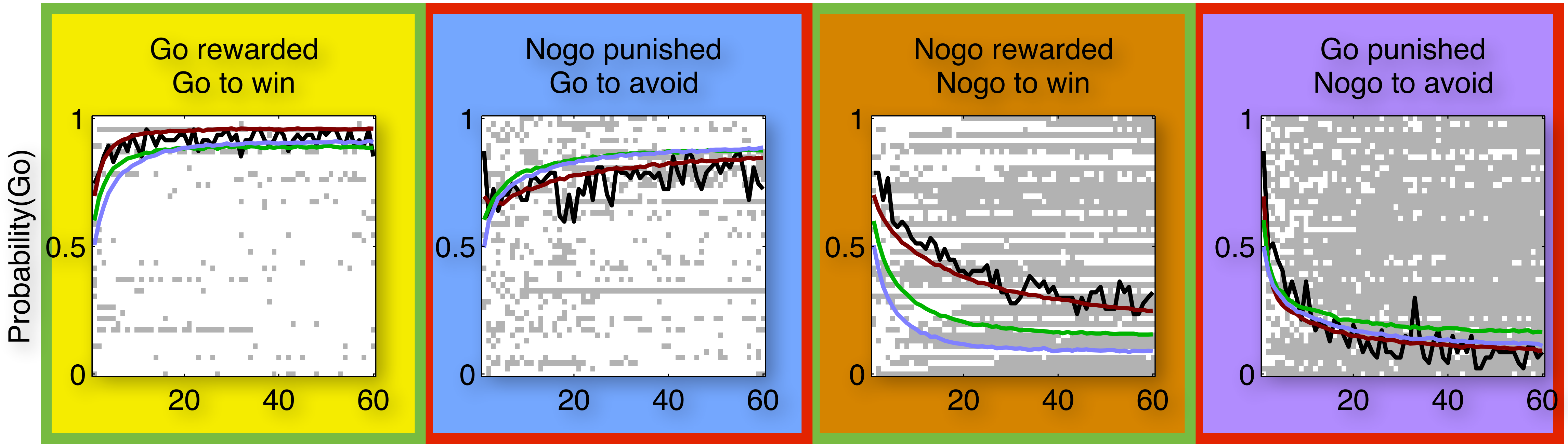
Integrate out your parameters

Model comparison: overfitting?





Note: same number of parameters



Behavioural data modelling

- Are no panacea
  - statistics about specific aspects of decision machinery
  - only account for part of the variance
- Model needs to match experiment
  - ensure subjects actually do the task the way you wrote it in the model
  - model comparison
- Model = Quantitative hypothesis
  - strong test
  - need to compare **models**, not **parameters**
  - includes all consequences of a hypothesis for choice

Thanks

- Peter Dayan
- Daniel Schad
- Nathaniel Daw

- SNSF FNSNF Swiss National Science Foundation
- DFG DFG Deutsche Forschungsgemeinschaft