Reinforcement Learning
Crash course

Quentin Huys

Wellcome Trust Centre for Neuroimaging
Gatsby Computational Neuroscience Unit
Medical School
UCL

Magdeburg University, June 20th 2009

Overview

- RL Crash course
- Some behavioural considerations
- Fitting behaviour with RL models

Types of models

- phenomenological
  - what?
  - summarise and describe data
    - mean
    - correlations, fMRI
- mechanistic
  - how?
  - algorhitmic
- normative
  - why?
  - teleological, notions of optimality

Types of models

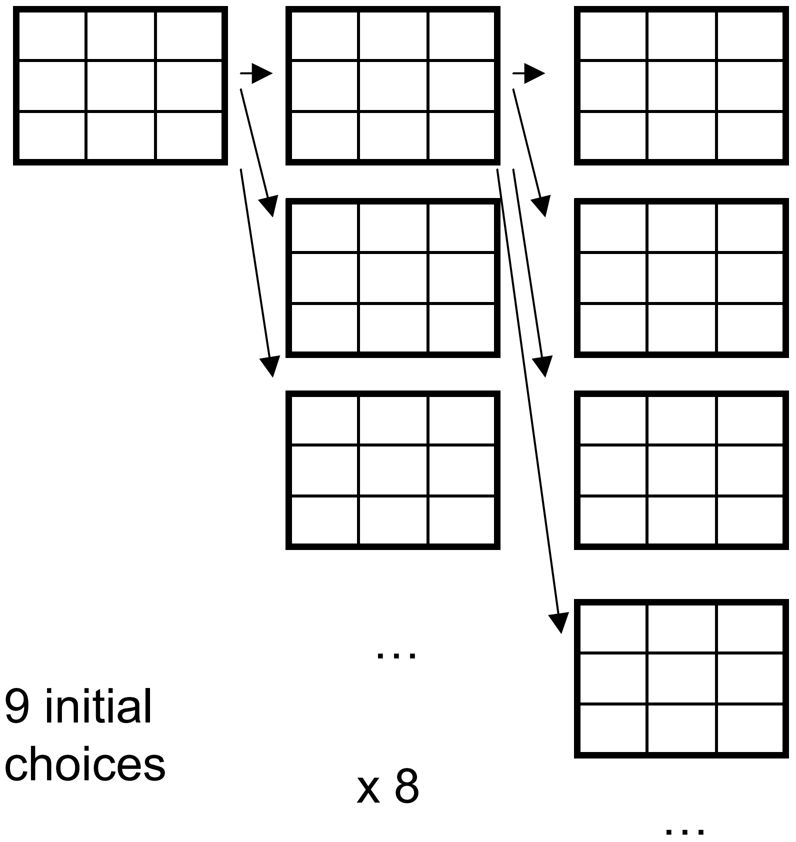
- mechanistic
  - how?
  - algorhitmic
- normative
  - why?
  - teleological, notions of optimality

Types of models

- normative
  - why?
  - teleological, notions of optimality

Types of models

Decisions: Let's play XOX



9 initial choices

(really just 3)

x 8

(really just 5)

x 7

Can go through all possible board settings

9! to 230 symmetries etc.

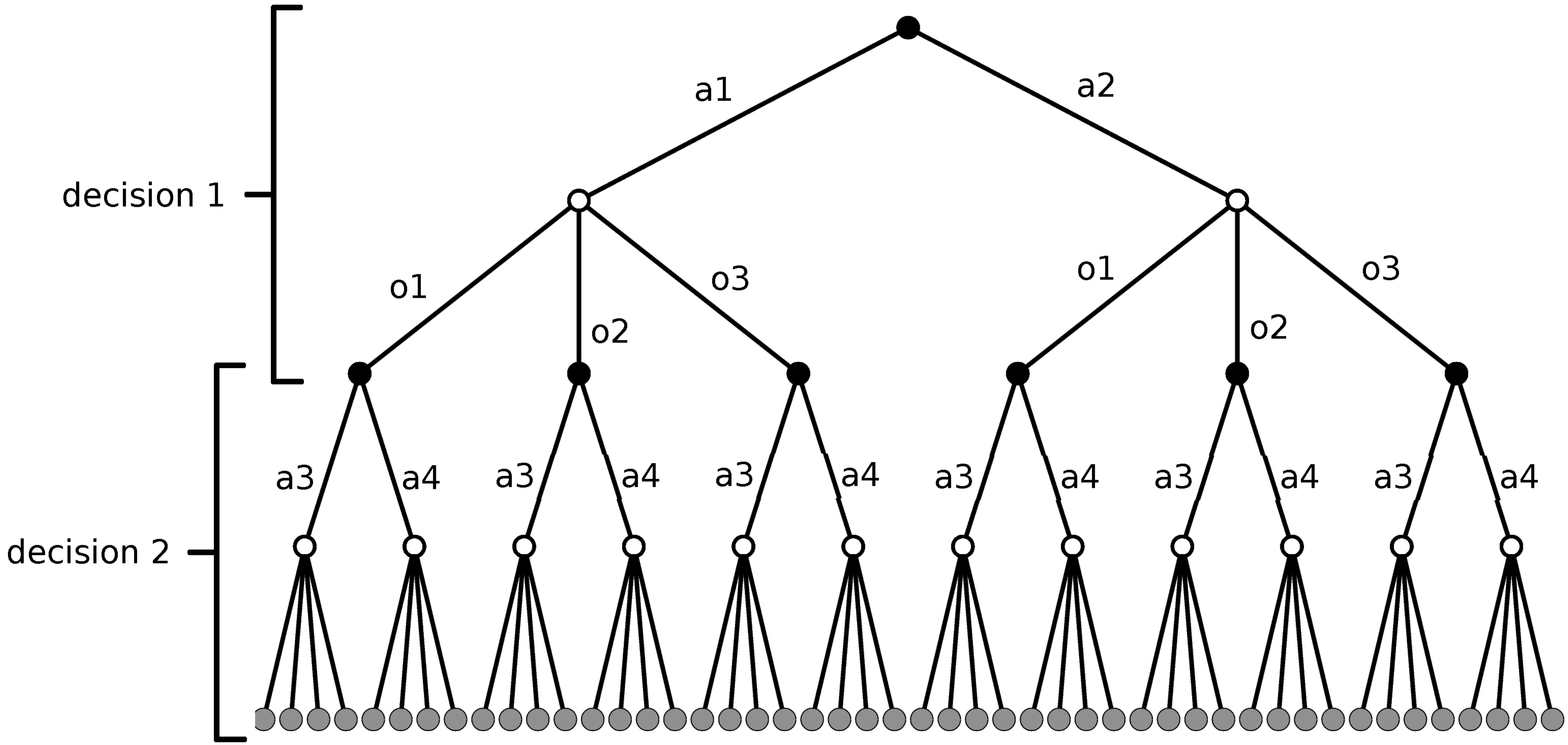
For each, consider all following positions

Chose move that gets you closest to winning or keeps you furthest from losing (minimax/maximin)

Choose best sequence in advance:

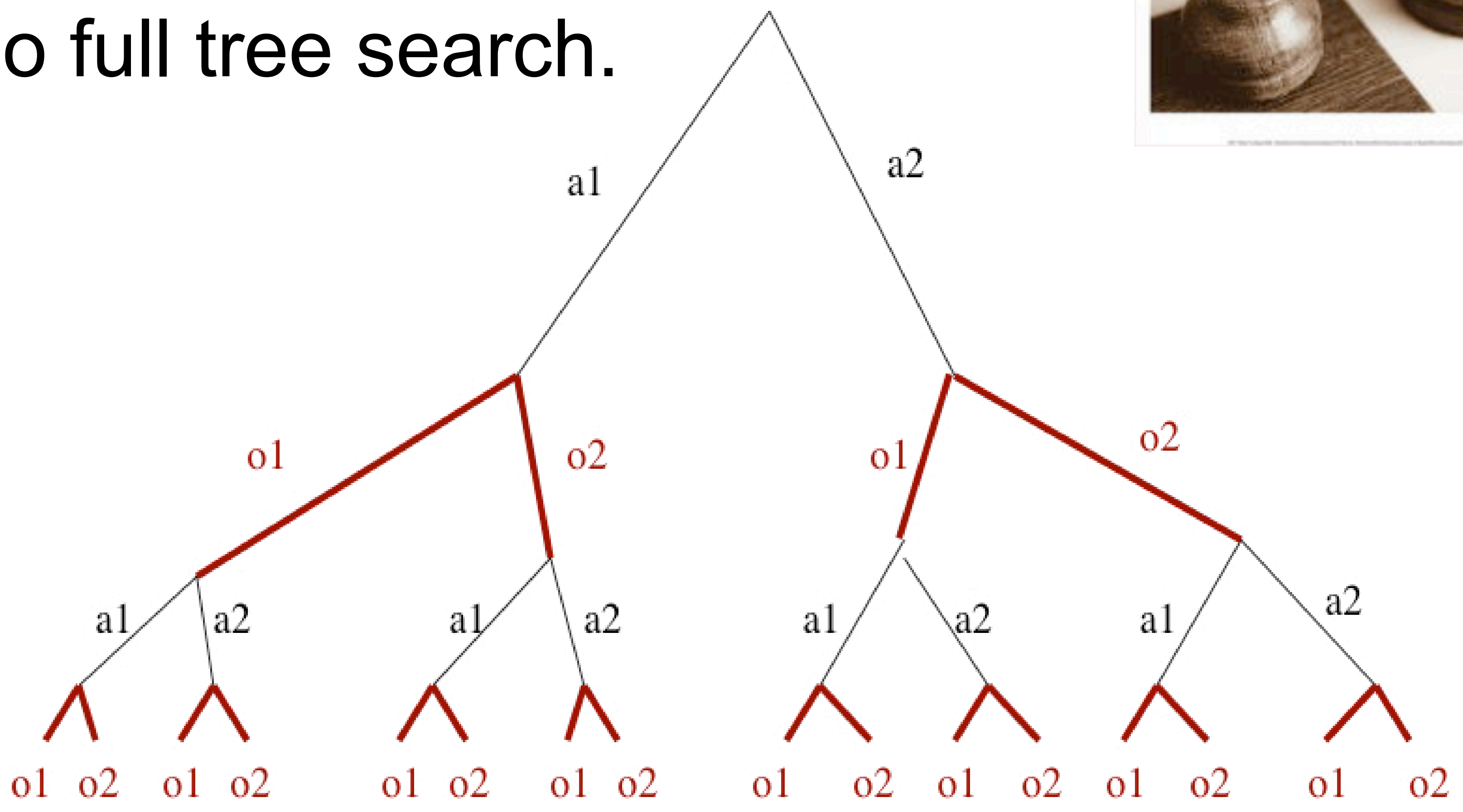
$$\{a_t\} \leftarrow \underset{\{a_t\}}{\operatorname{argmax}} \sum_{t=1}^{\infty} r_t$$

Processing depth



Chess

- Each move 30 odd choices
- 30^{40}?
- MANY!!!
  - Legal boards ~10^{123}
- Can't just do full tree search.



Soooo....?

Soooo....?

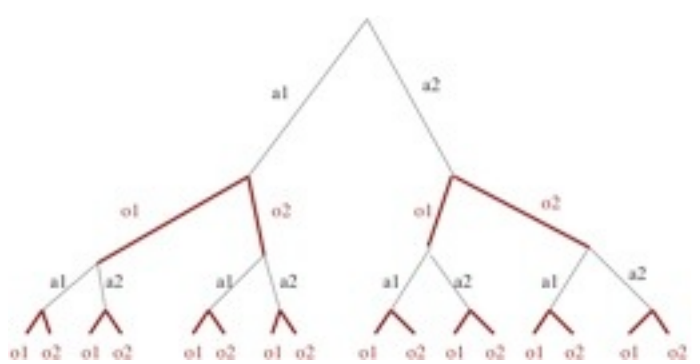
How do players do it?
How did Deep Blue beat Kasparov?

Multiple, parallel, decision-making systems

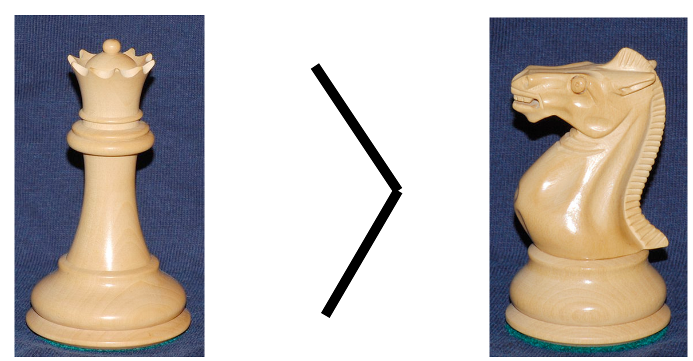
Multiple decision systems "Controllers"

Competition and collaboration

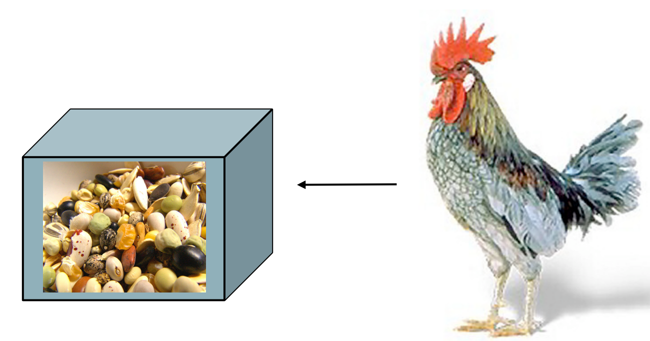
Goal-directed system
Tree search



Habit system
Experience average



Innate system
Evolutionary strategy



In humans, animals and computers...

Setup

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

After Sutton and Barto 1998

Discounting

- Why discount?

$$\sum_{t=0}^{\infty} r_t = \infty \qquad \text{if no absorbing state}$$

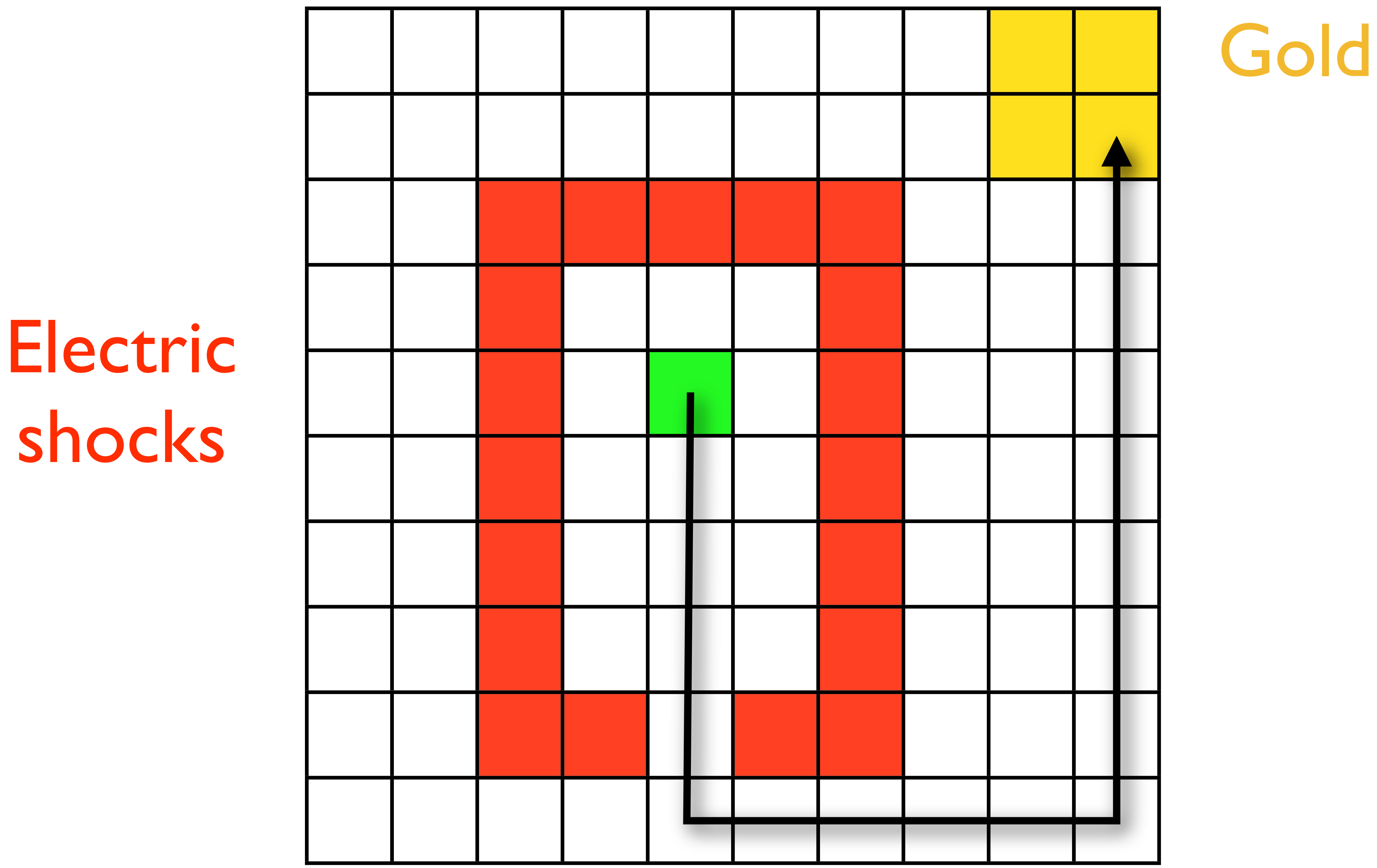
- When discount?
  - infinite horizons

$$\sum_{t=0}^{\infty} \gamma^t r_t < \infty \qquad \text{for most r of interest}$$

  - finite, exponentially distributed horizons

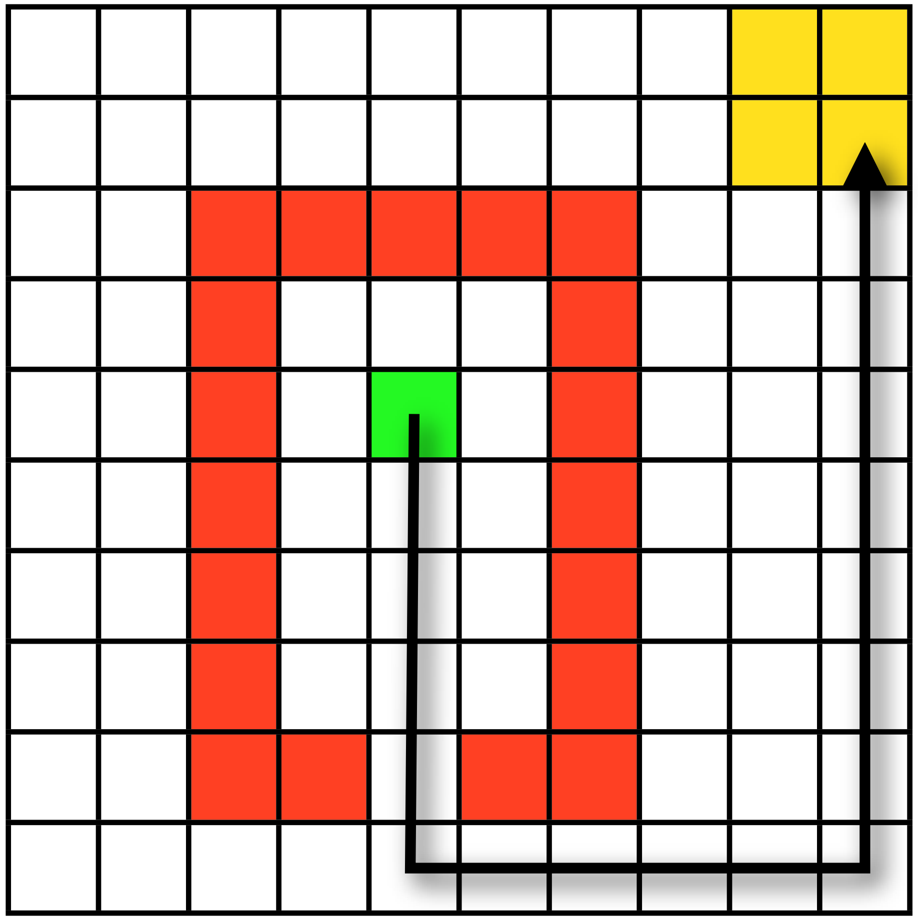
$$\sum_{t=0}^{T} \gamma^t r_t \qquad T \sim \frac{1}{\tau} e^{t/\tau}$$

State space



A Markov Decision Problem

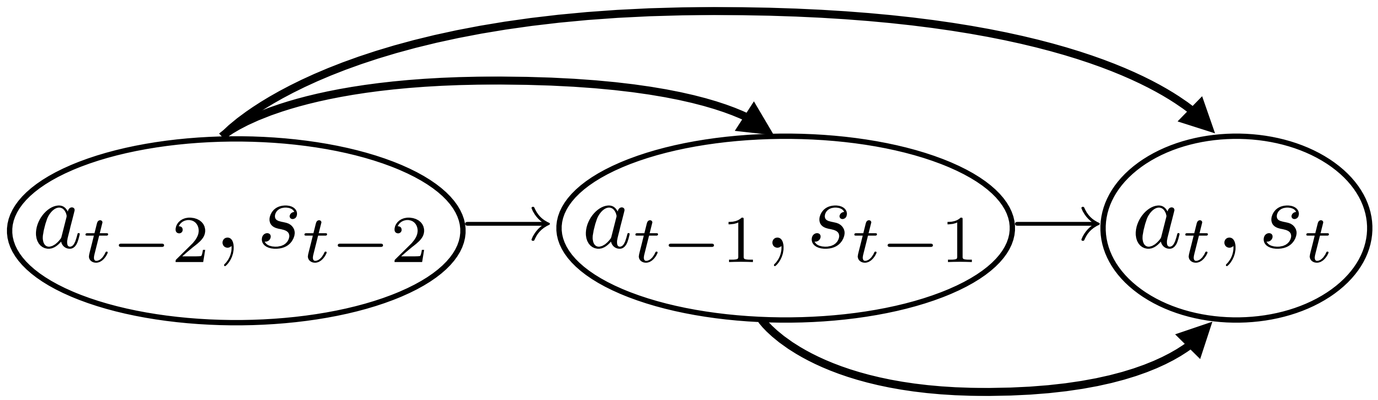
$$
\begin{aligned}
\boxed{s_t \quad &\in \quad \mathcal{S}} \\
a_t \quad &\in \quad \mathcal{A} \\
\mathcal{T}^a_{ss'} \quad &= \quad p(s_{t+1}|s_t, a_t) \\
r_t \quad &\sim \quad \mathcal{R}(s_{t+1}, a_t, s_t) \\
\pi(a|s) \quad &= \quad p(a|s)
\end{aligned}
$$



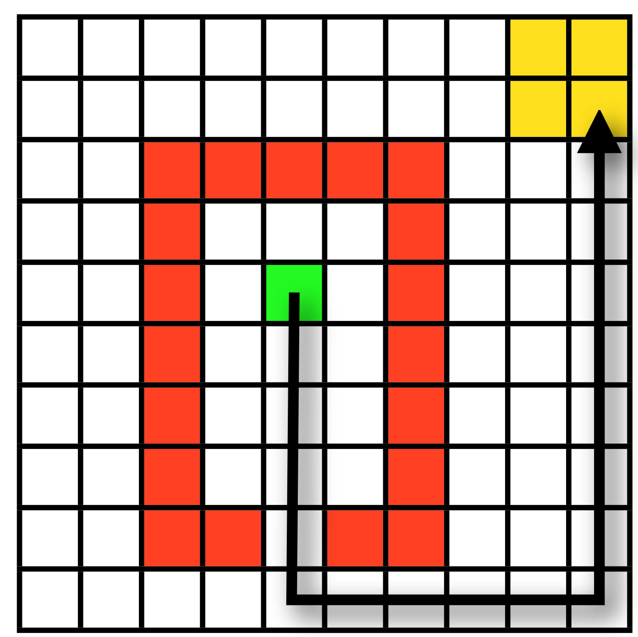
Markovian!

Markov state-space descriptions

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



Velocity

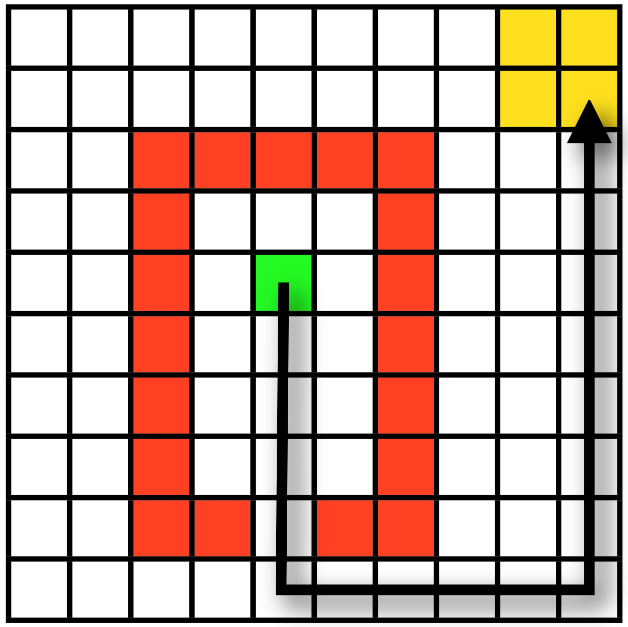


Markov state-space descriptions

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

$$a_{t-2}, s_{t-2} \longrightarrow a_{t-1}, s_{t-1} \longrightarrow a_t, s_t$$

Velocity



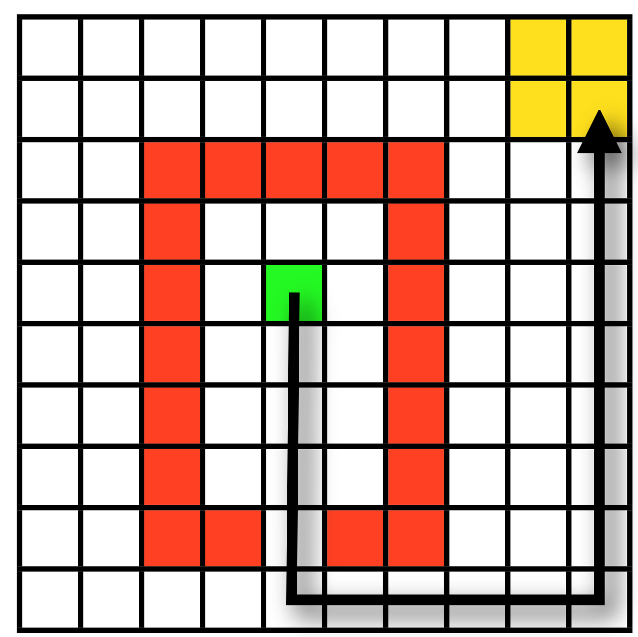
Markov state-space descriptions

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

$$a_{t-2}, s_{t-2} \longrightarrow a_{t-1}, s_{t-1} \longrightarrow a_t, s_t$$

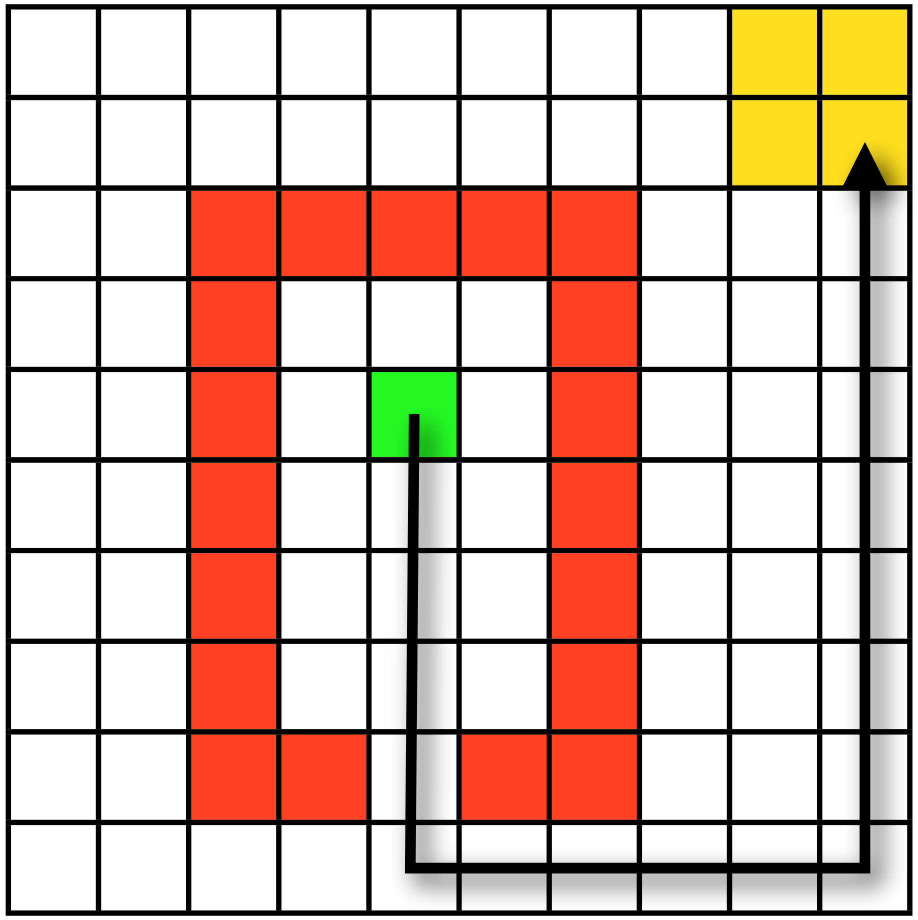
Velocity

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



MDP

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



Rewards

- Any outcome we want to maximise

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

- Rewards & punishments
  - reward = - punishment

- Matching

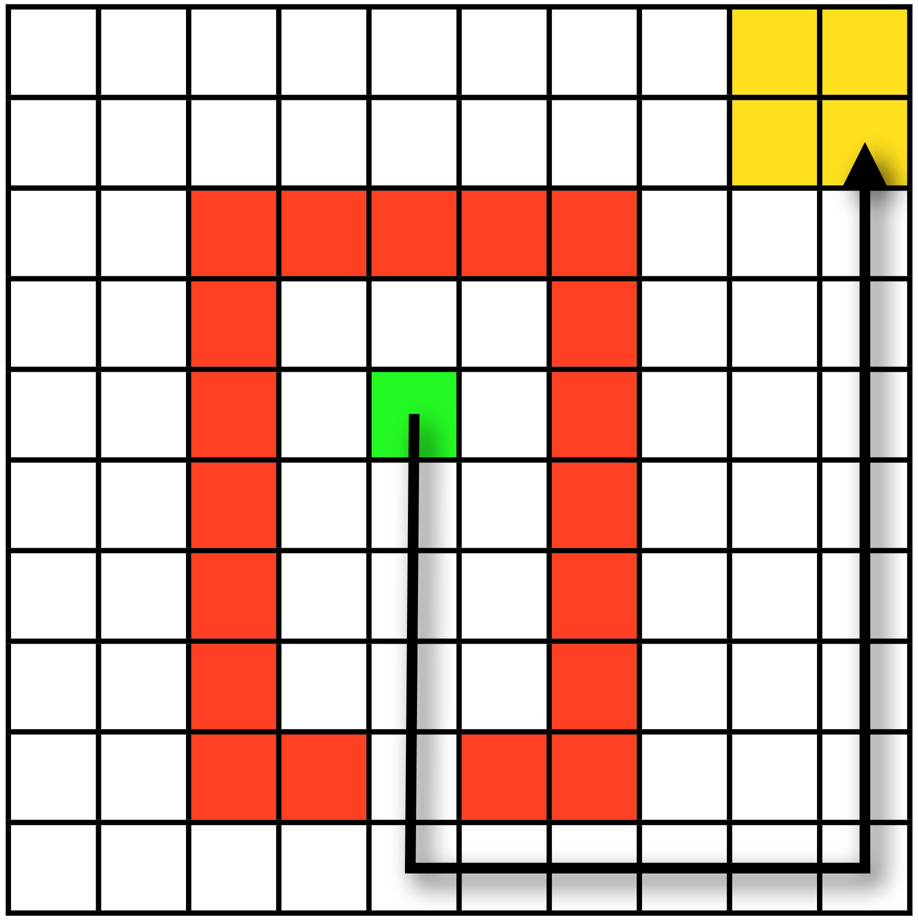
$$p(a_t) \propto E\left[\sum_t r_t | a_t\right]$$

- Revealed preferences $p(a_t) \rightarrow \mathcal{R}?$
  - Ryanair?
- Discounting

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

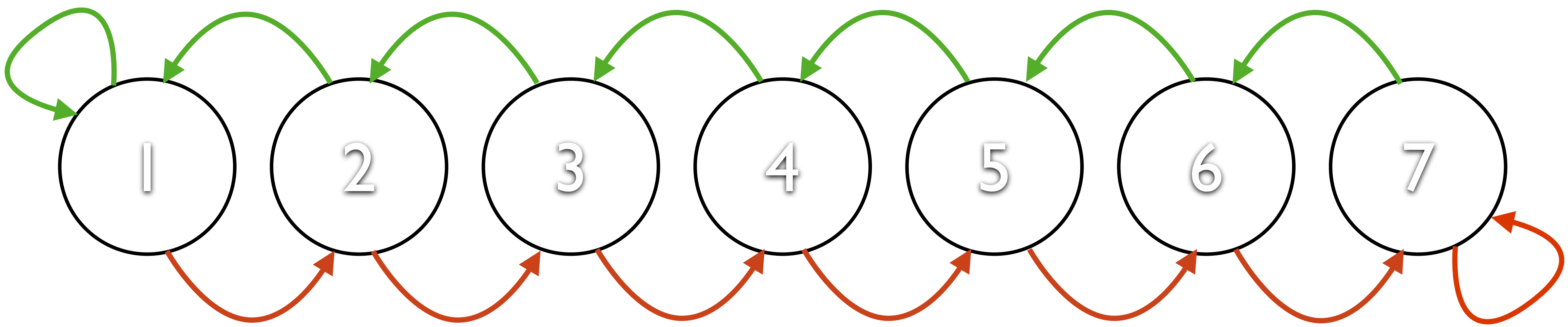
MDP

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



Actions

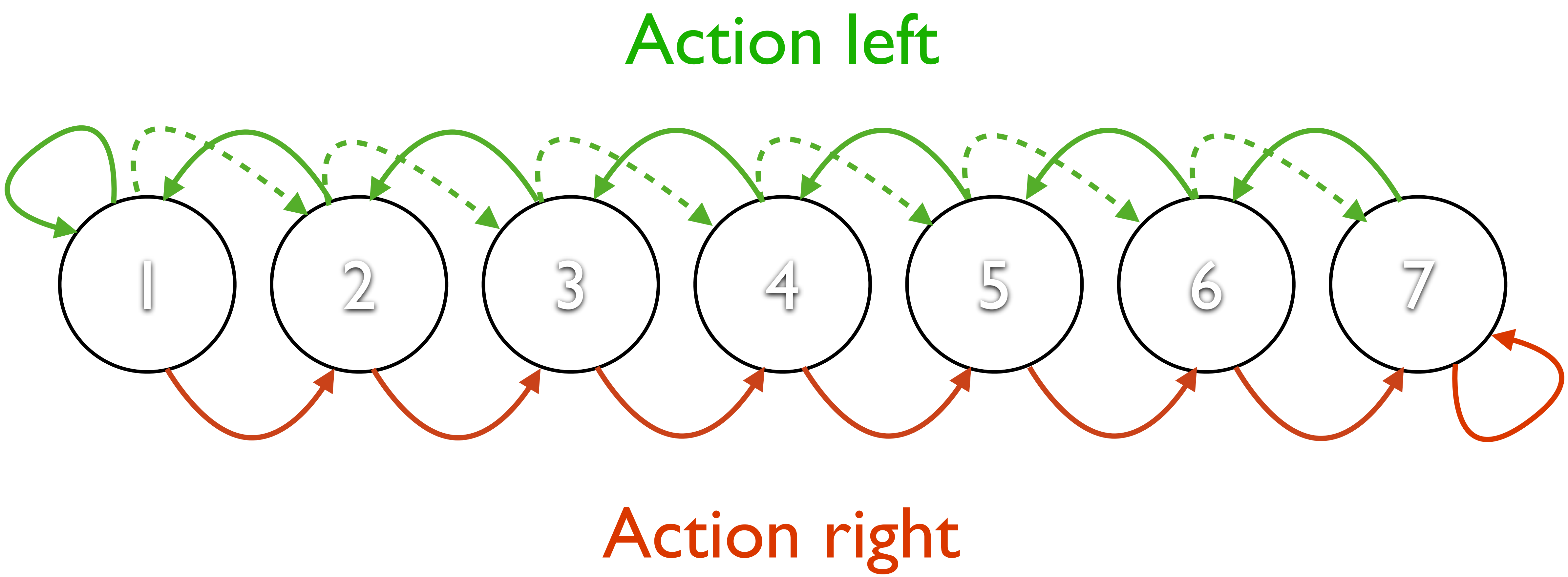
Action left



Action right

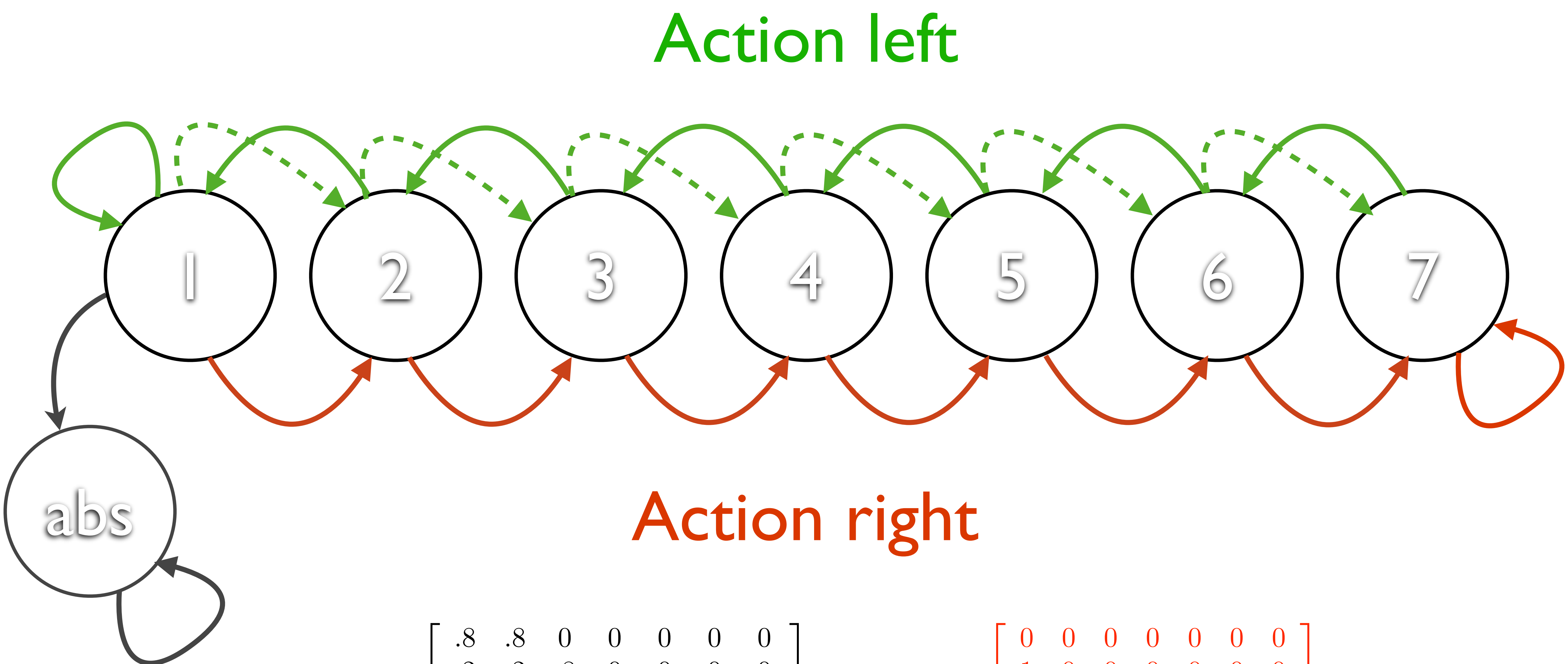
$$T^{\text{left}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad T^{\text{right}} = \begin{bmatrix} 0 & 0 & 0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

Actions



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

Actions

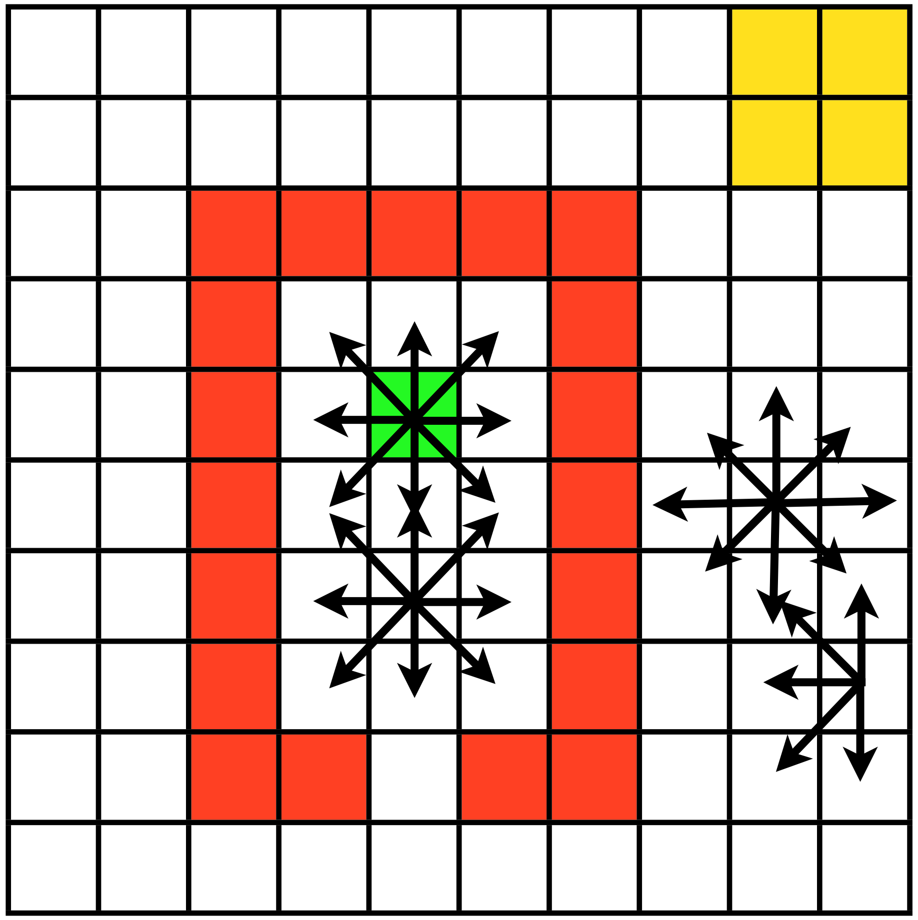


$$T^{\text{left}} = \begin{bmatrix} .8 & .8 & 0 & 0 & 0 & 0 & 0 \\ .2 & .2 & .8 & 0 & 0 & 0 & 0 \\ 0 & 0 & .2 & .8 & 0 & 0 & 0 \\ 0 & 0 & 0 & .2 & .8 & 0 & 0 \\ 0 & 0 & 0 & 0 & .2 & .8 & 0 \\ 0 & 0 & 0 & 0 & 0 & .2 & .8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad T^{\text{right}} = \begin{bmatrix}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

Absorbing state -> max eigenvalue < 1

MDP

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



MDP

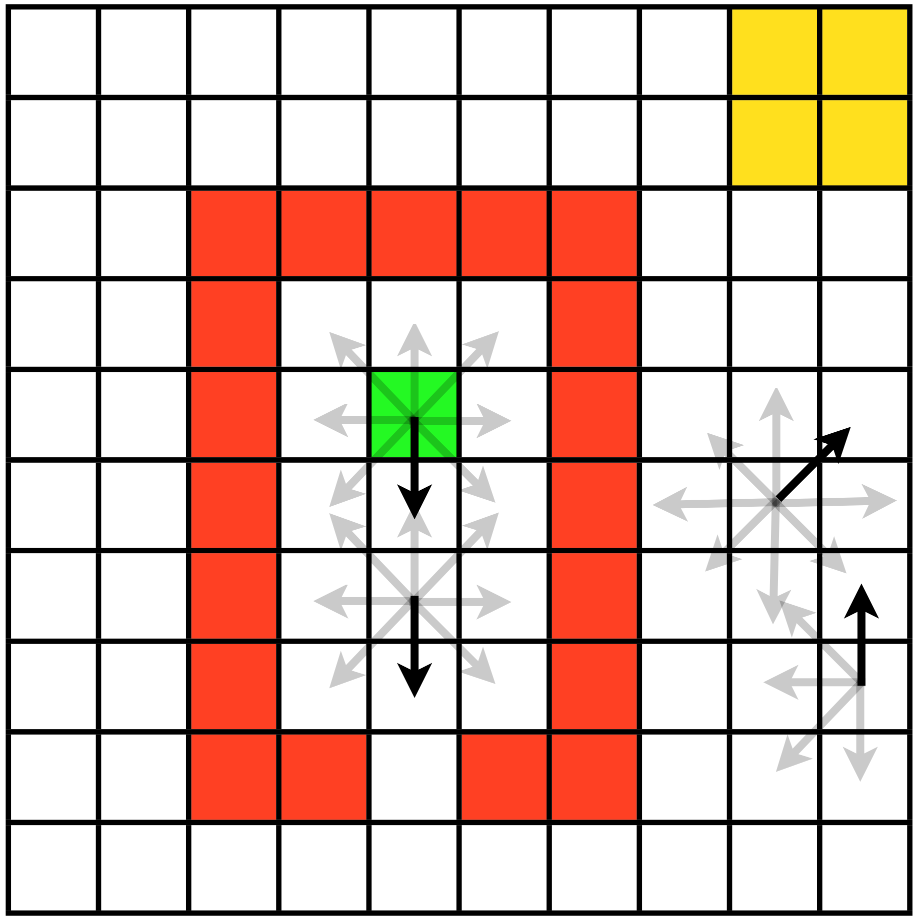
$$s_t \in \mathcal{S}$$

$$a_t \in \mathcal{A}$$

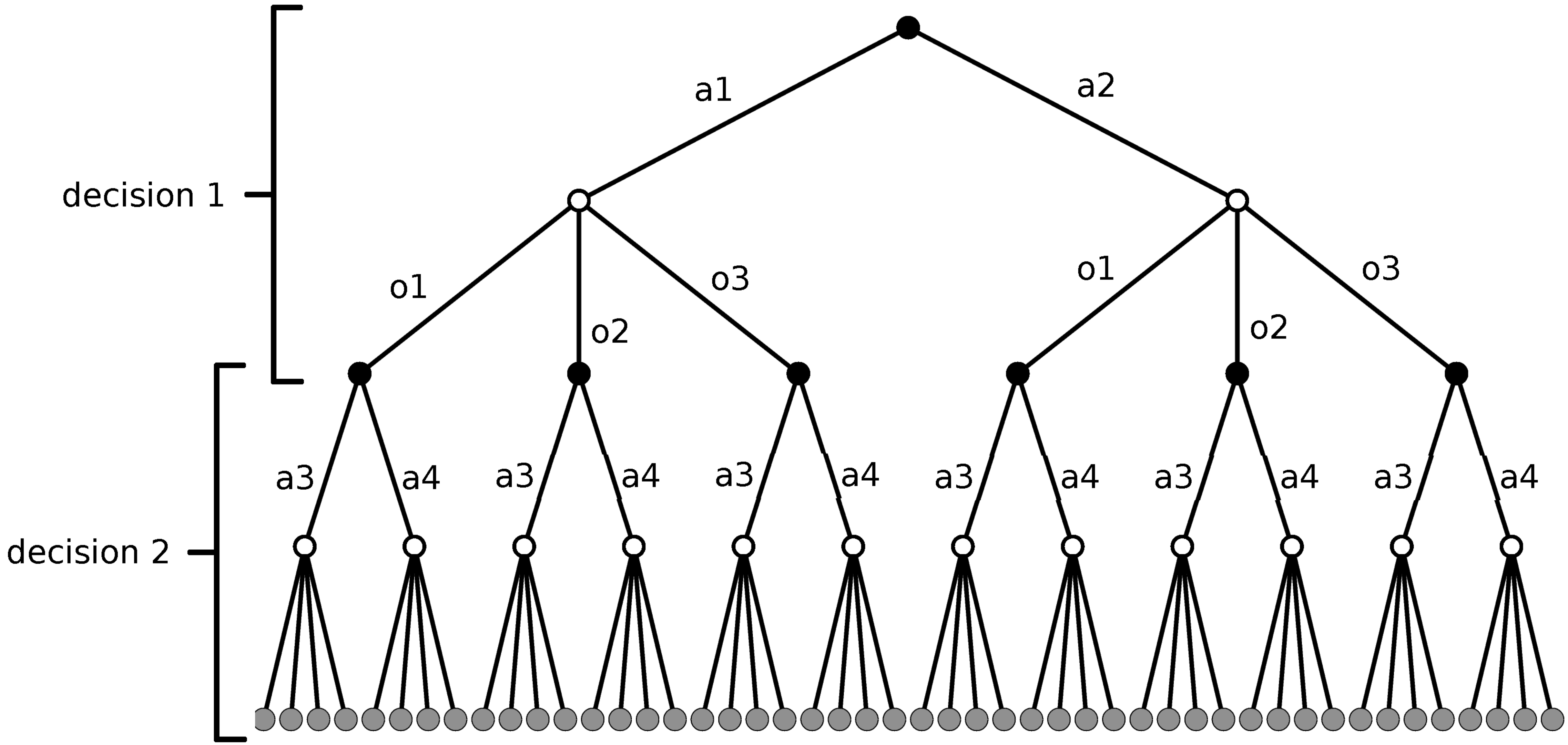
$$\mathcal{T}^a_{ss'} = p(s_{t+1}|s_t, a_t)$$

$$r_t \sim \mathcal{R}(s_{t+1}, a_t, s_t)$$

$$\boxed{\pi(a|s) = p(a|s)}$$

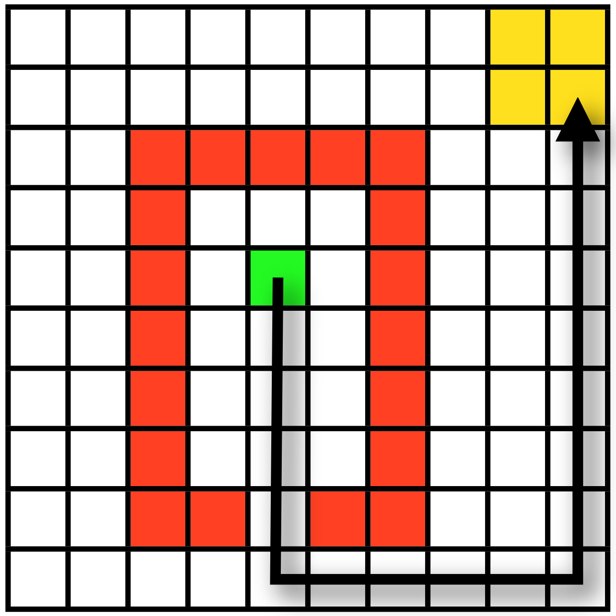


Decision tree: exhaustive search



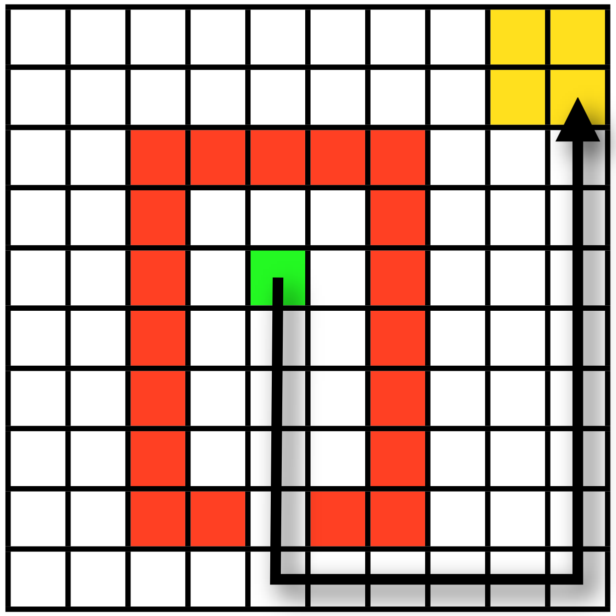
w^d

Markov Decision Problems



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

Markov Decision Problems



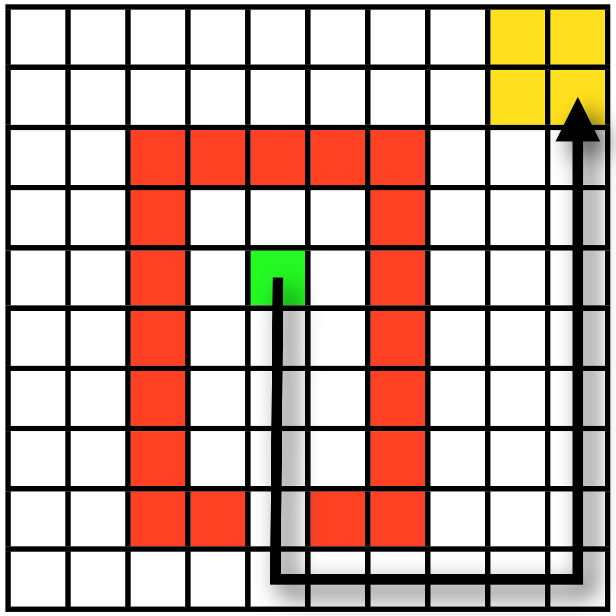
$$
\begin{aligned}
V(s_t) &= \boxed{\mathbb{E}\left[r_1 \middle| s_t = s\right]} + \mathbb{E}\left[V(s_{t+1})\right] \\
r_1 &\sim \mathcal{R}(s_2, a_1, s_1) \\
\mathbb{E}\left[r_1 | s_t = s\right] &= \mathbb{E}\left[\sum_{s_{t+1}} p(s_{t+1}|s_t, a_t)\mathcal{R}(s_{t+1}, a_t, s_t)\right] \\
&= \sum_{a_t} p(a_t|s_t)\left[\sum_{s_{t+1}} p(s_{t+1}|s_t, a_t)\mathcal{R}(s_{t+1}, a_t, s_t)\right] \\
&= \sum_{a_t} \pi(a_t, s_t)\left[\sum_{s_{t+1}} \mathcal{T}^{a_t}_{s_t s_{t+1}} \mathcal{R}(s_{t+1}, a_t, s_t)\right]
\end{aligned}
$$

Bellman equation

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

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

$$\mathbb{E}\left[V(s_{t+1})\right] = \sum_a \pi(a, s_t)\left[\sum_{s_{t+1}} \mathcal{T}^a_{s_t s_{t+1}} V(s_{t+1})\right]$$



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

Bellman Equation

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

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

Q values

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

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

$$= \mathbb{E}\left[\sum_{t=1}^{\infty} r_t | s, a \right]$$

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

Bellman Equation

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

$$\frac{1}{|\mathcal{S}|} \sum_{a,s,s'} \mathbf{1}(\mathcal{T}^a_{ss'} > 0)$$

Solving the Bellman Equation

Option 1: turn it into update equation

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

Option 2: linear solution (w/ absorbing states)

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

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:

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

We can update the policy:

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

Or all at once:

$$V^{\pi_{i+1}}(s) = \max_a \sum_{s'} \mathcal{T}^a_{ss'} \left[\mathcal{R}^a_{ss} + V^{\pi_i}(s')\right]$$

Policy iteration

Policy evaluation

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

Policy update

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

Policy iteration

Policy evaluation

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

Value iteration

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

Policy update

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

Solving the Bellman Equation

Option 3: sampling

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

Solving the Bellman Equation

Option 3: sampling

$$V(s) = \int da\, \pi(a,s) \left[\int ds'\, \mathcal{T}^{a}_{ss'} \left[\mathcal{R}(s', a, s) + V(s') \right] \right]$$

Solving the Bellman Equation

Option 3: sampling

$$V(s) = \int da\, \pi(a,s) \left[\int ds'\, \mathcal{T}^{a}_{ss'} \left[\mathcal{R}(s',a,s) + V(s') \right] \right]$$

Sampling:

Solving the Bellman Equation

Option 3: sampling

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

Sampling:

$$a = \int dx\, f(x) p(x)$$

Solving the Bellman Equation

Option 3: sampling

$$V(s) \quad = \quad \int da\, \pi(a,s) \left[\int ds'\, \mathcal{T}^{a}_{ss'} \left[\mathcal{R}(s',a,s) + V(s') \right] \right]$$

Sampling:

$$a \quad = \quad \int dx\, f(x) p(x)$$

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

Solving the Bellman Equation

Option 3: sampling

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

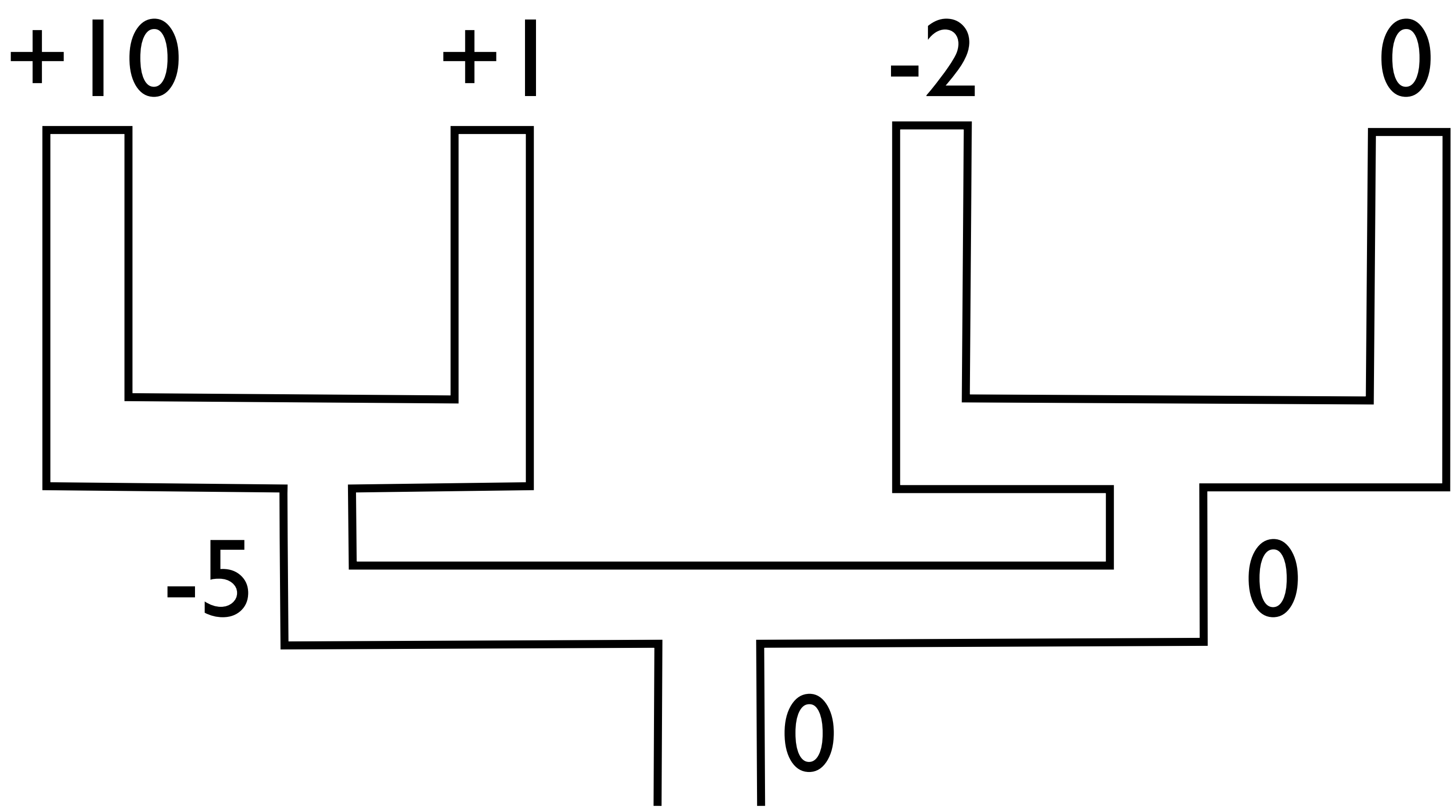
Sampling:

$$a = \int dx\, f(x) p(x)$$

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

$$x_i \sim q(x) \rightarrow \hat{a} = \frac{1}{N} \sum_i f(x_i) w_i \qquad \text{where} \quad w_i = \frac{p(x_i)}{q(x_i)}$$

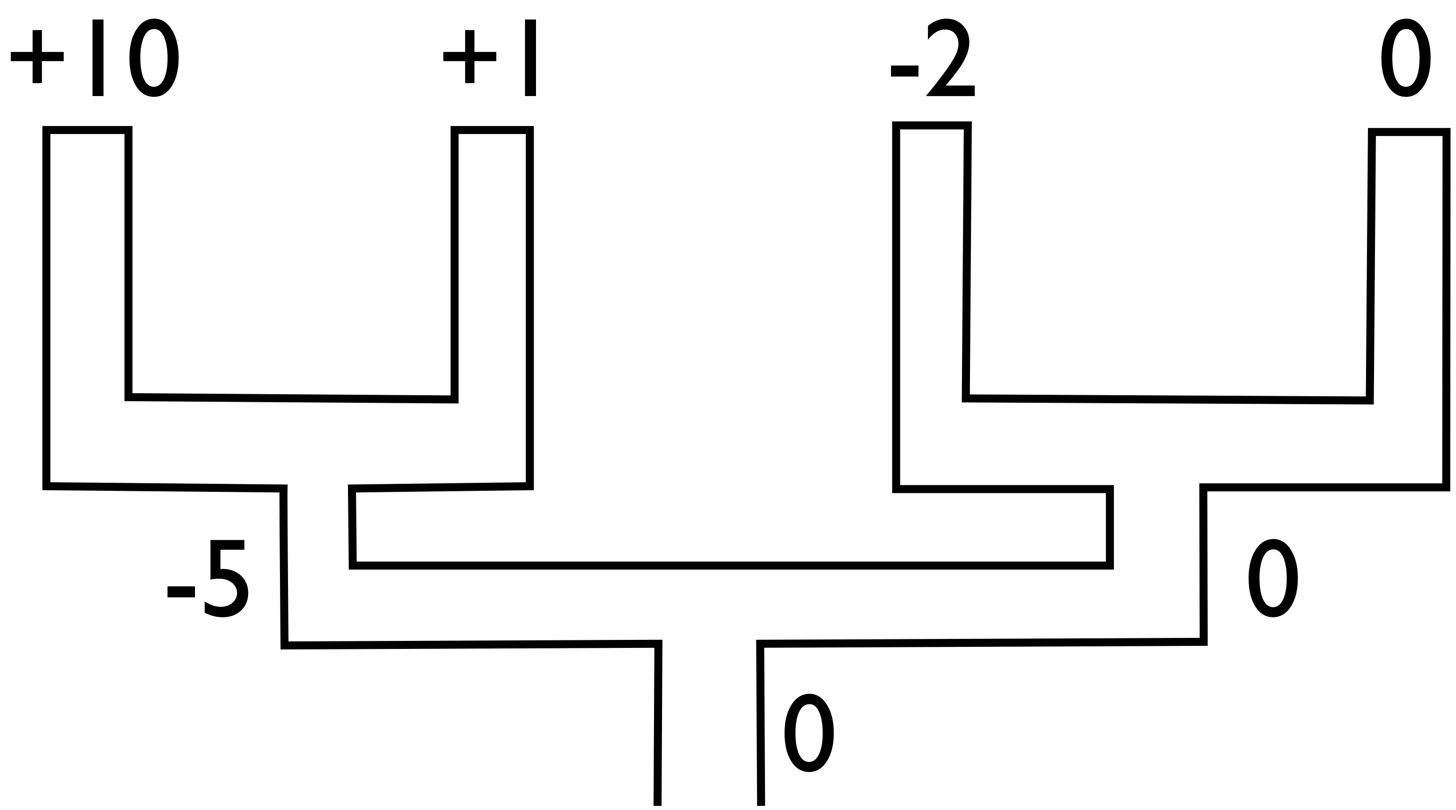
Model-free, Monte Carlo RL



Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r^i_{t'} \middle| s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL

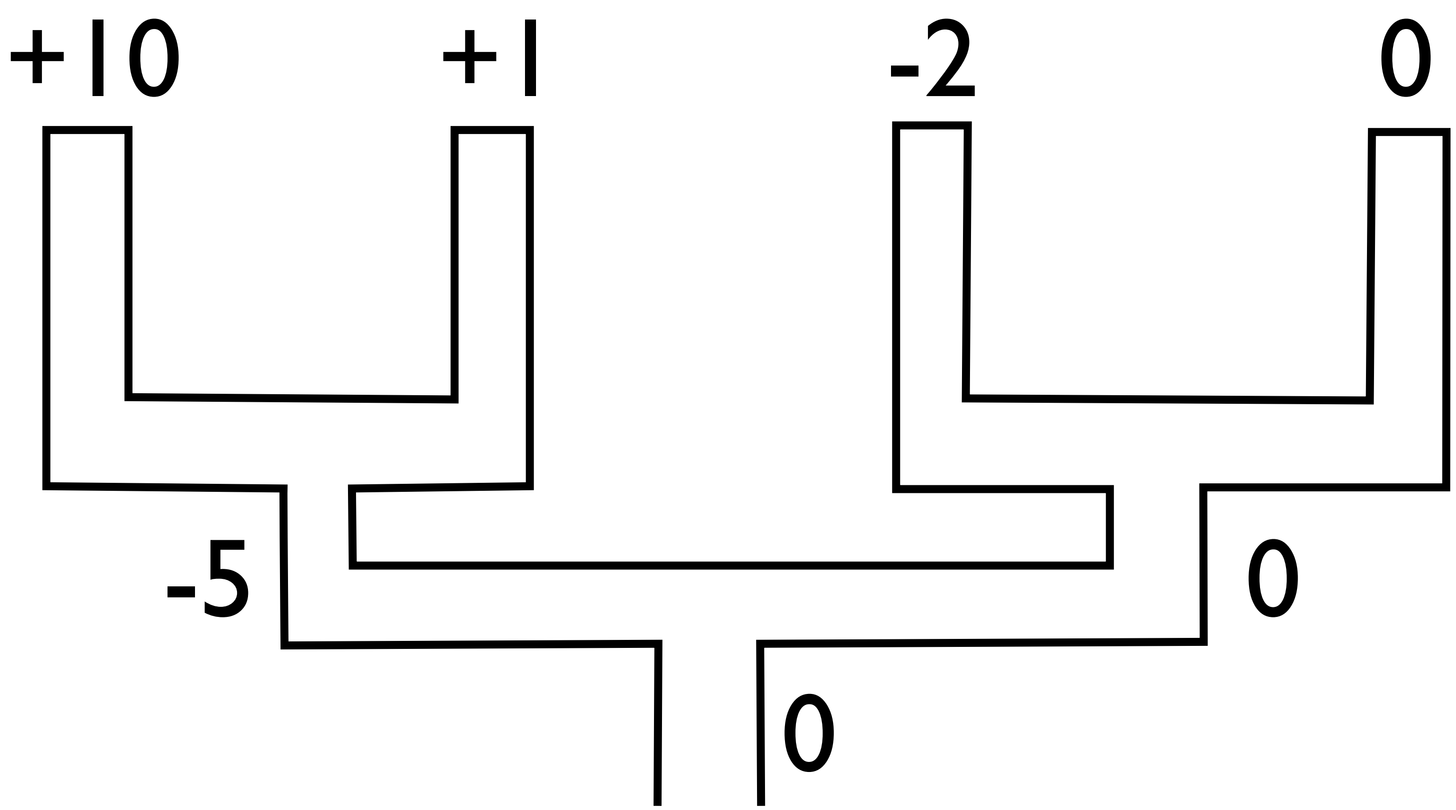


0L-5R1 = -4

Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r_{t'}^i | s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL

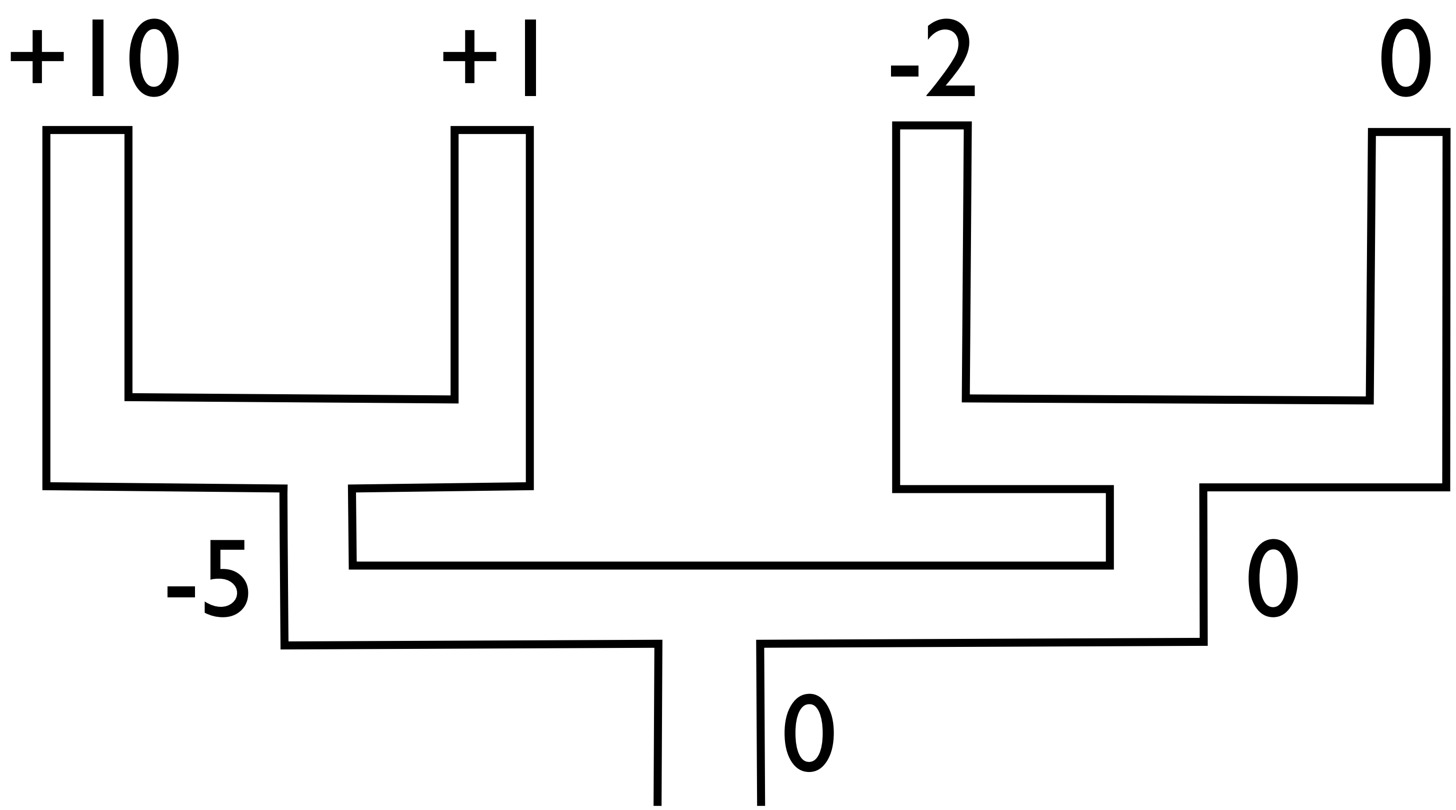


0L-5R1 = -4
0L-5R1 = -4

Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r^i_{t'} | s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL

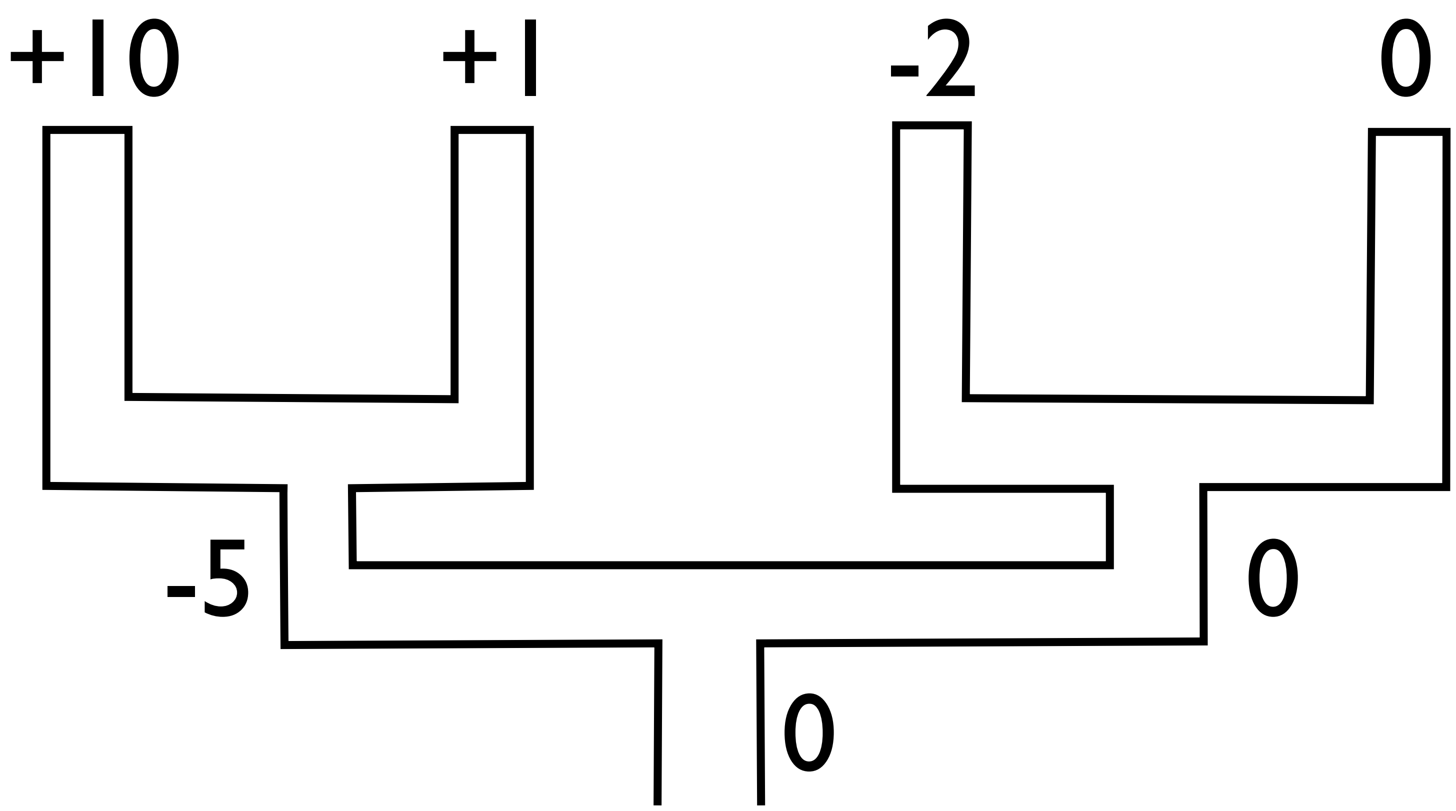


0L-5R1 = -4
0L-5R1 = -4
0R0R0 = 0

Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r^i_{t'} \middle| s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL

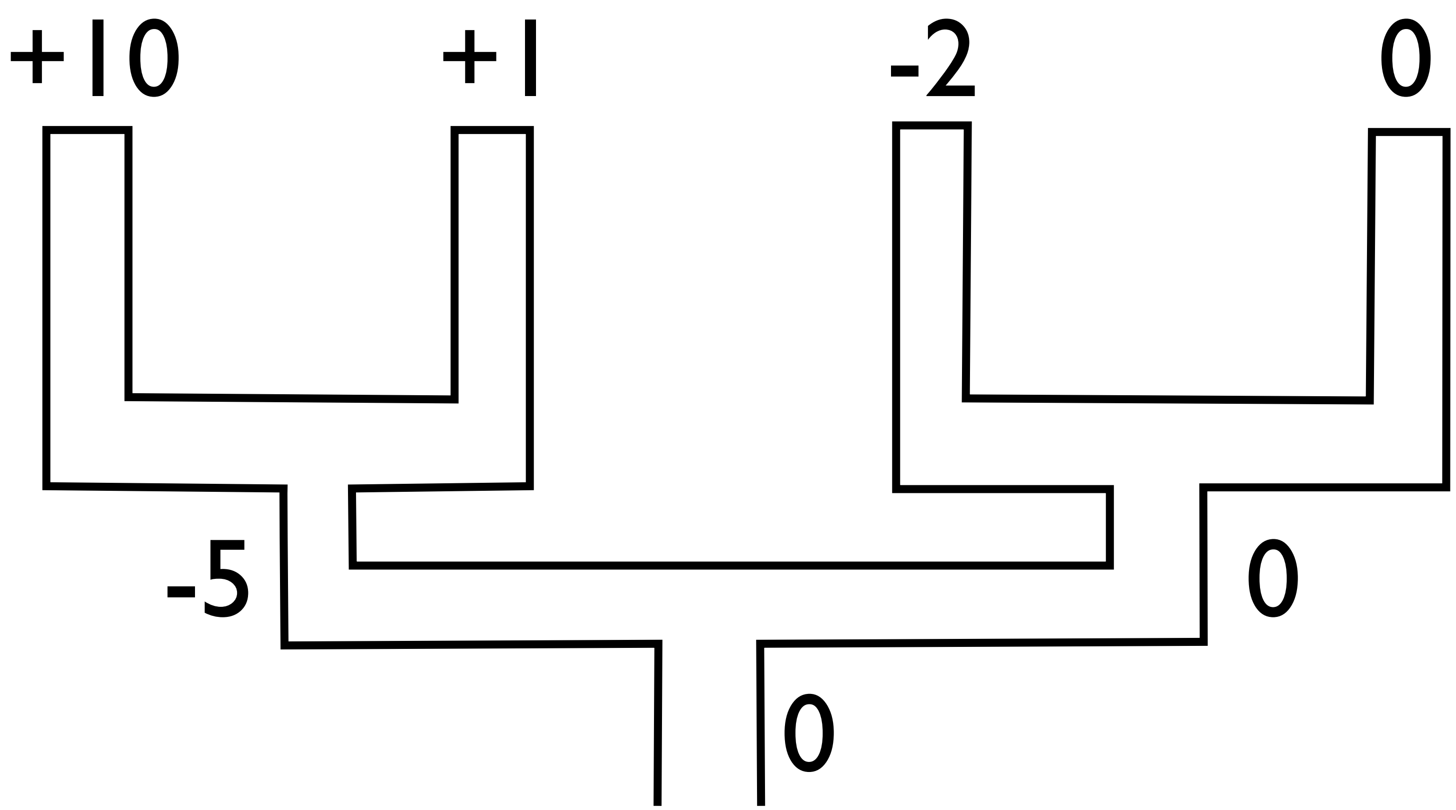


0L-5R1 = -4
0L-5R1 = -4
0R0R0 = 0
0R0L-2 = -2

Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r^i_{t'} | s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL

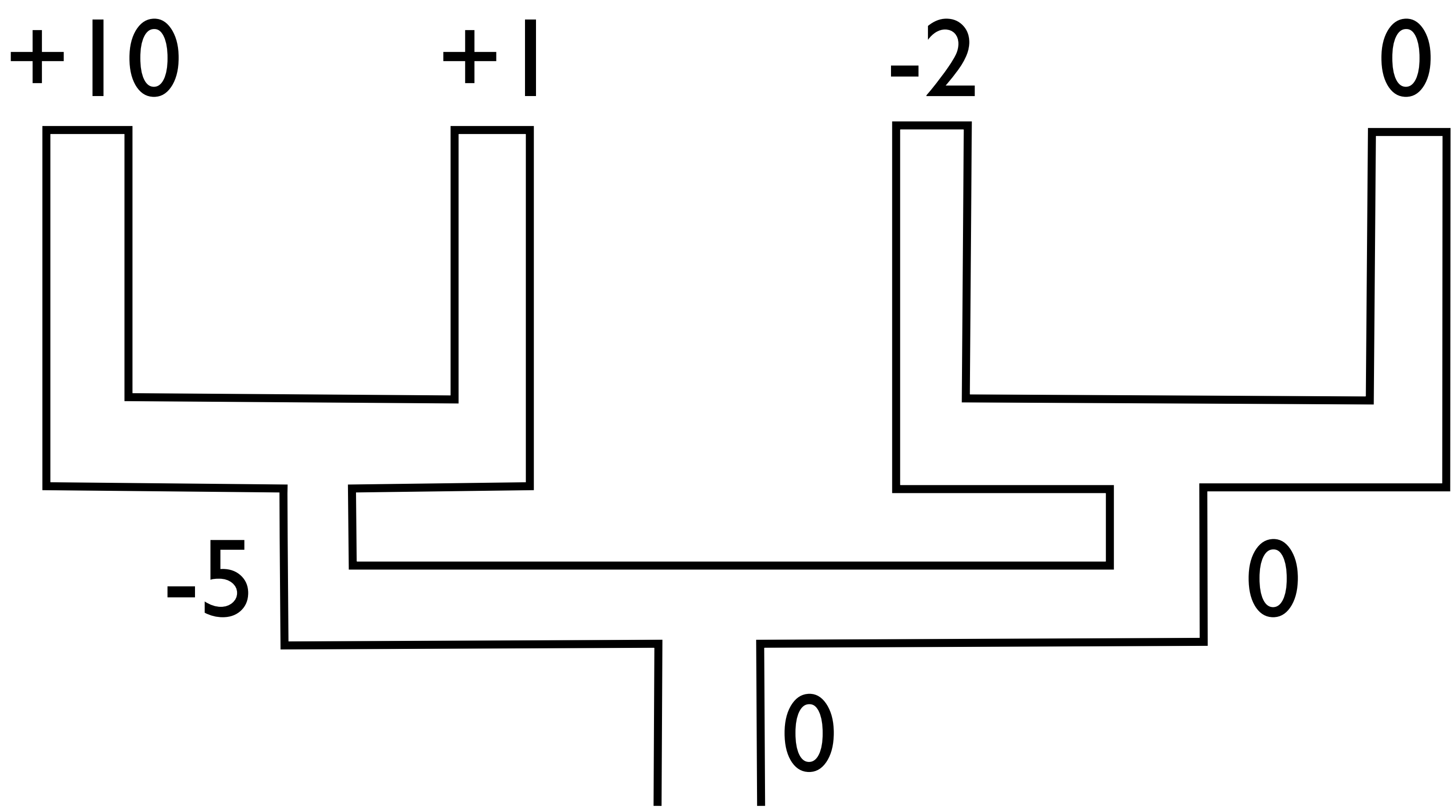


0L-5R1 = -4
0L-5R1 = -4
0R0R0 = 0
0R0L-2 = -2
0L-5L10= 5

Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r^i_{t'} | s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL

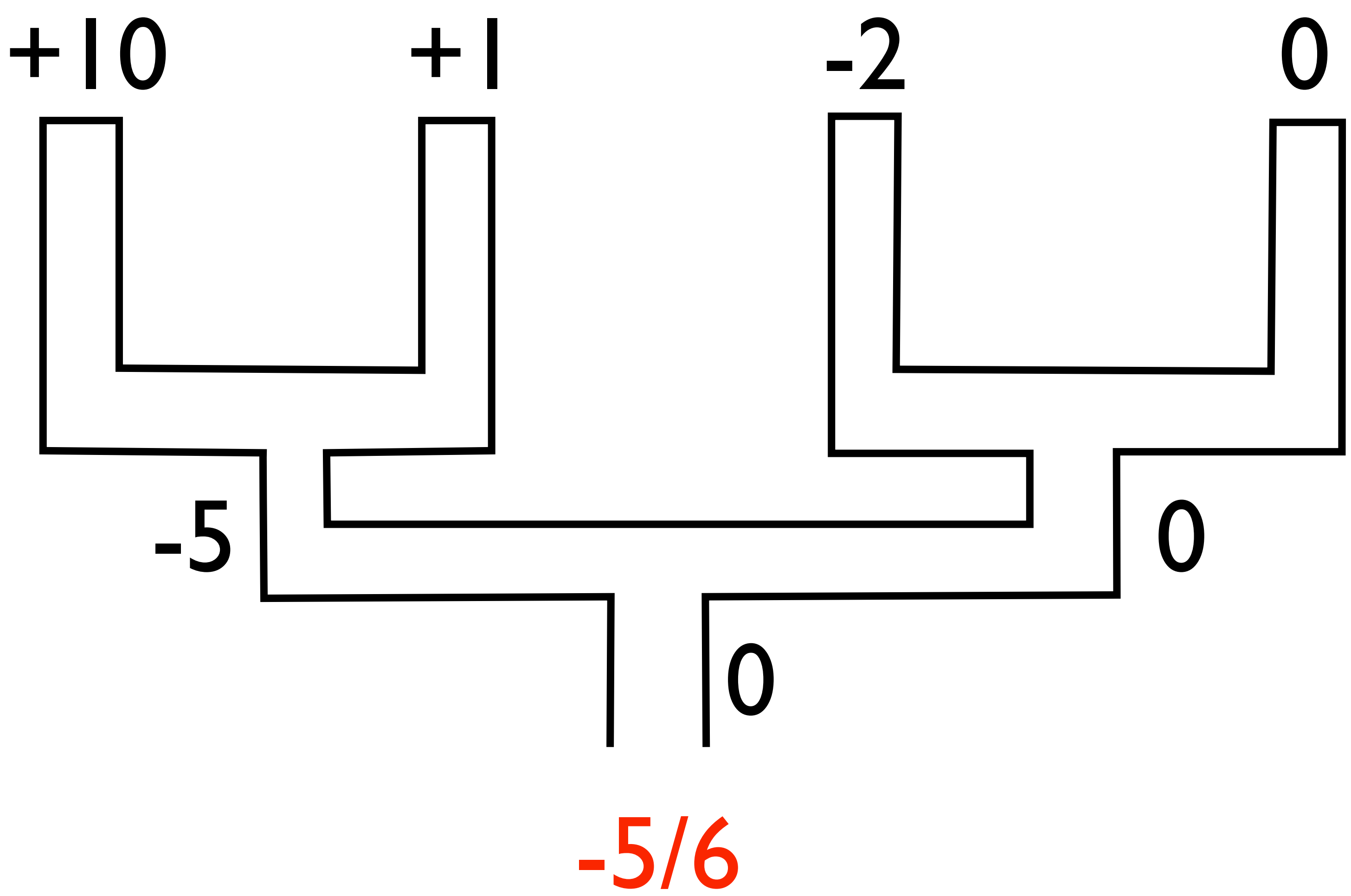


0L-5R1 = -4
0L-5R1 = -4
0R0R0 = 0
0R0L-2 = -2
0L-5L10= 5
0R0R0 = 0

Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r_{t'}^{i} | s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL

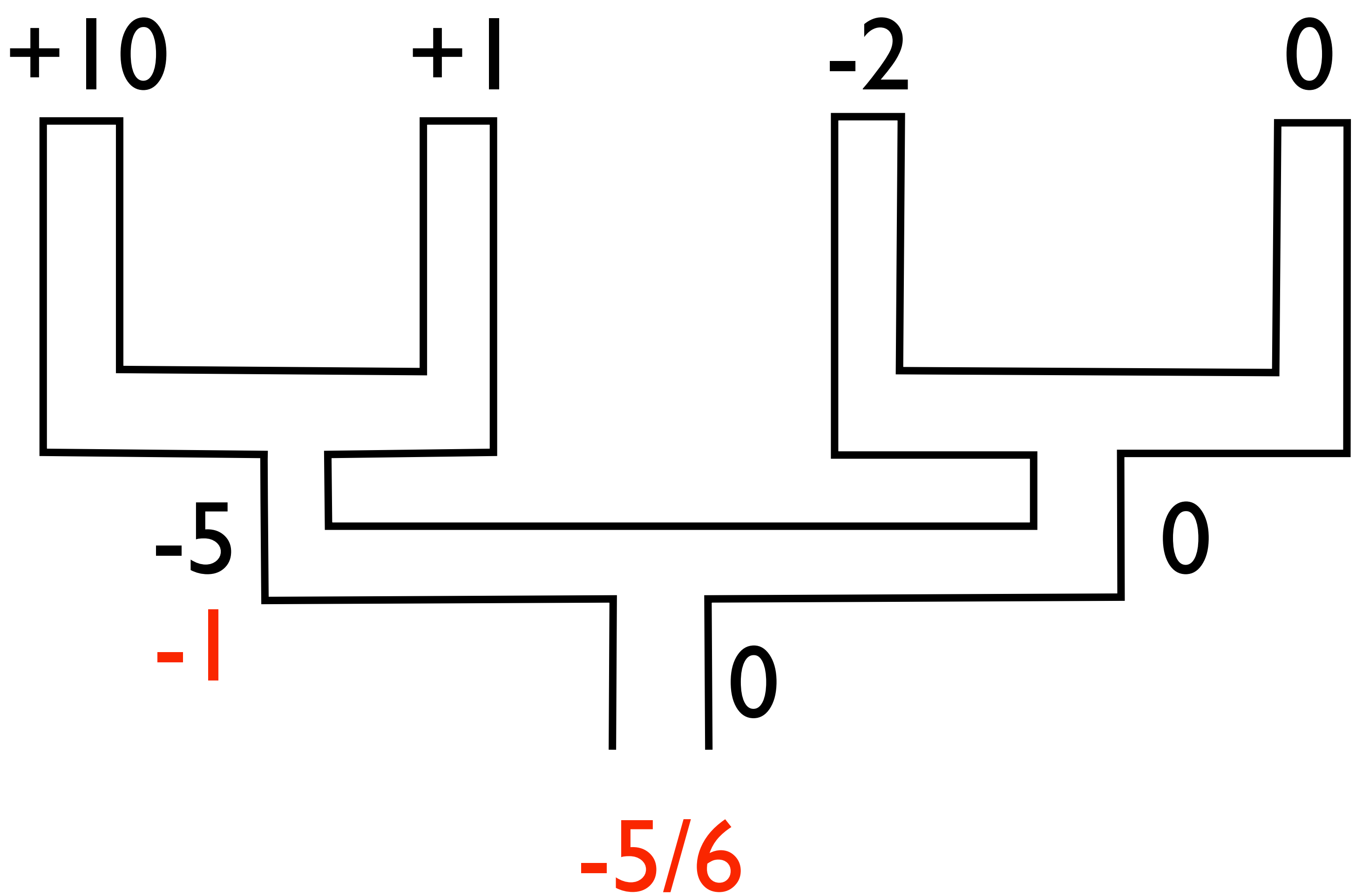


0L-5R1 = -4
0L-5R1 = -4
0R0R0 = 0
0R0L-2 = -2
0L-5L10= 5
0R0R0 = 0

Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r^i_{t'} | s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL

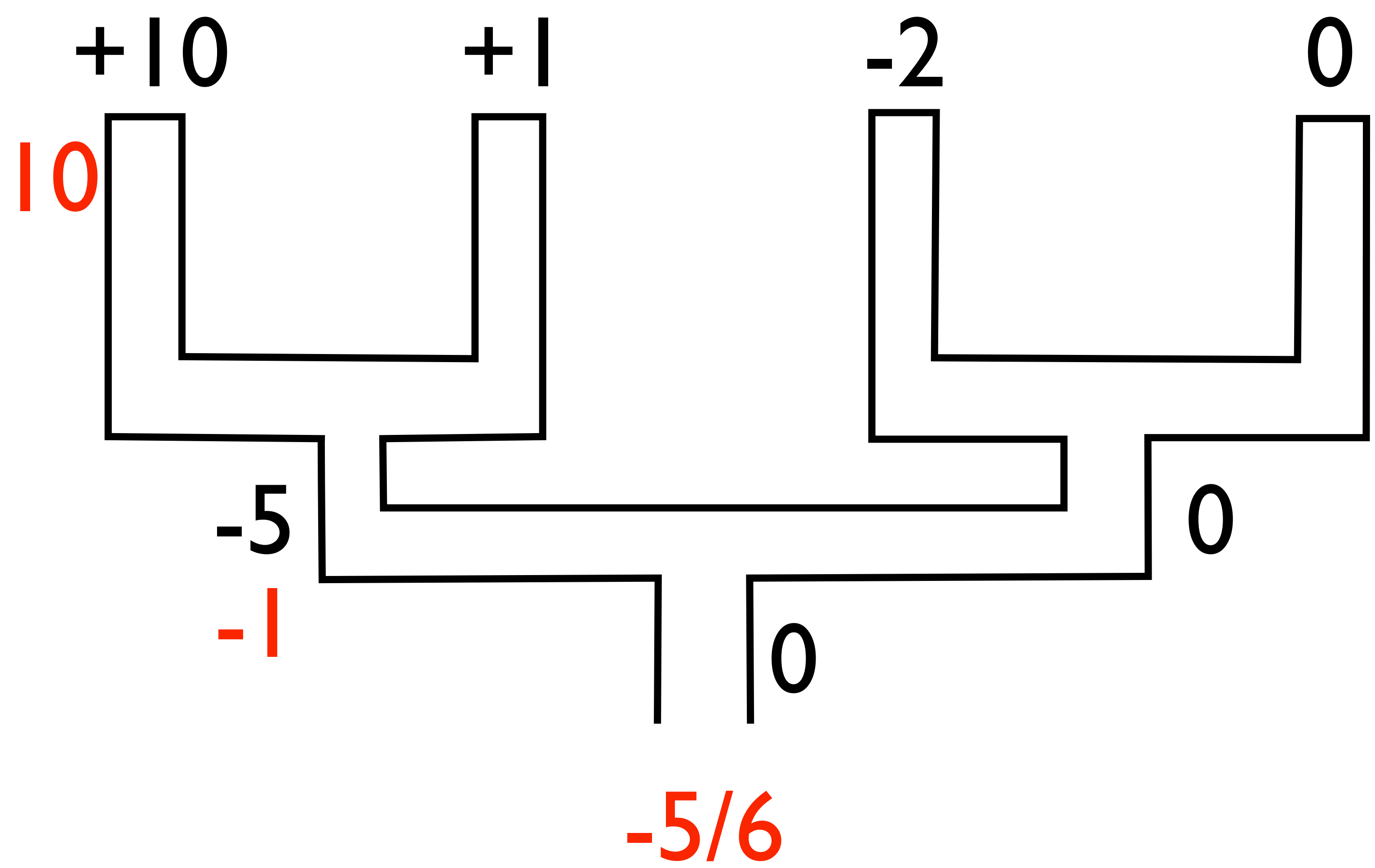


0L-5R1 = -4
0L-5R1 = -4
0R0R0 = 0
0R0L-2 = -2
0L-5L10= 5
0R0R0 = 0

Or rather, learn state-action values directly:

$$\mathcal{Q}(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r^i_{t'} | s_0 = s, a_0 = a \right\}$$

Model-free, Monte Carlo RL



0L-5R1 = -4

0L-5R1 = -4

0R0R0 = 0

0R0L-2 = -2

0L-5L10 = 5

0R0R0 = 0

Or rather, learn state-action values directly:

$$Q(s,a) = \frac{1}{N}\sum_i \left\{ \sum_{t'=1}^{T} r^i_{t'} | s_0 = s, a_0 = a \right\}$$

Probabilistic policies

- softmax

$$p(a|s) = \frac{e^{\beta \mathcal{Q}(s,a)}}{\sum_{a'} e^{\beta \mathcal{Q}(s,a')}}$$

  - β trades off exploration vs exploitation

- ε-greedy:

$$p(a|s) = \begin{cases} 1-\epsilon & \text{if } a = a^* \\ \epsilon & \text{else} \end{cases}$$

  - ε trades off exploration vs exploitation

- When should policy be updated?

Monte Carlo RL

- Average over sample state paths
- No knowledge of transitions T or rewards R
  - No model of the world!
  - But need to sample from it
- standard deviation ~ $\frac{1}{\sqrt{N}}$
  - values policy-dependent
    - importance sampling
  - Sample relevant state-actions
- Curse of dimensionality
  - hurts sampling
- exploration / exploitation?

0L-5R1 = -4
0L-5R1 = -4
0R0R0 = 0
0R0L-2 = -2
0L-5L10= 5
0R0R0 = 0

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

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

Model-free RL: TD learning

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

Model-free RL: 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

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

Model-free RL: 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

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

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

Model-free RL: 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

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

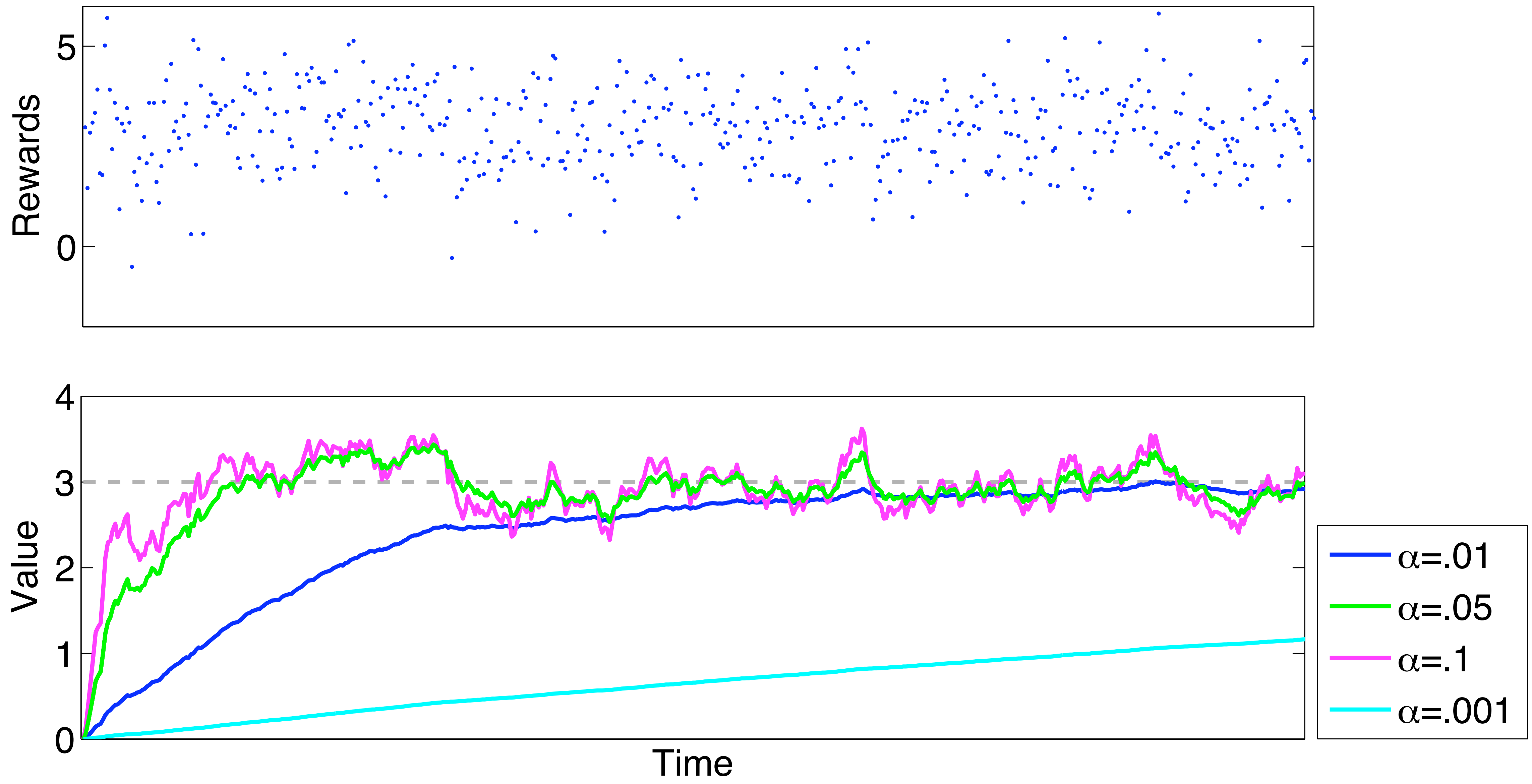
$$\delta_t = -V_t(s_t) + r_t + V_t(s_{t+1})$$

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

Learning rate

$$
\begin{aligned}
V_{t+1}(s) &= V_t(s) + \alpha\delta_t \\
&= V_t(s) + \alpha(r_t - V_t(s)) \\
&= (1-\alpha)V_t(s) + \alpha r_t \\
&= (1-\alpha)^2 V_{t-1}(s) + \alpha[(1-\alpha)r_{t-1} + r_t] \\
&= (1-\alpha)^t V_0(s) + \alpha \sum_{t'=1}^{t} (1-\alpha)^{t-t'} r_{t'}
\end{aligned}
$$

Fixed learning rate



Fixed learning rate = exponential forgetting

Assumption of changing world

TD learning

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

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

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

$$V_{t+1}(s) = (1-\alpha)V_t(s) + \alpha(V_t(s_{t+1}) + r_t)$$

Model-free: TD vs Markov

B1	
B1	
B1	
B1	
B1	
B1	
B0	
A0	B0

Markov

V(A)=0

V(B)=3/4

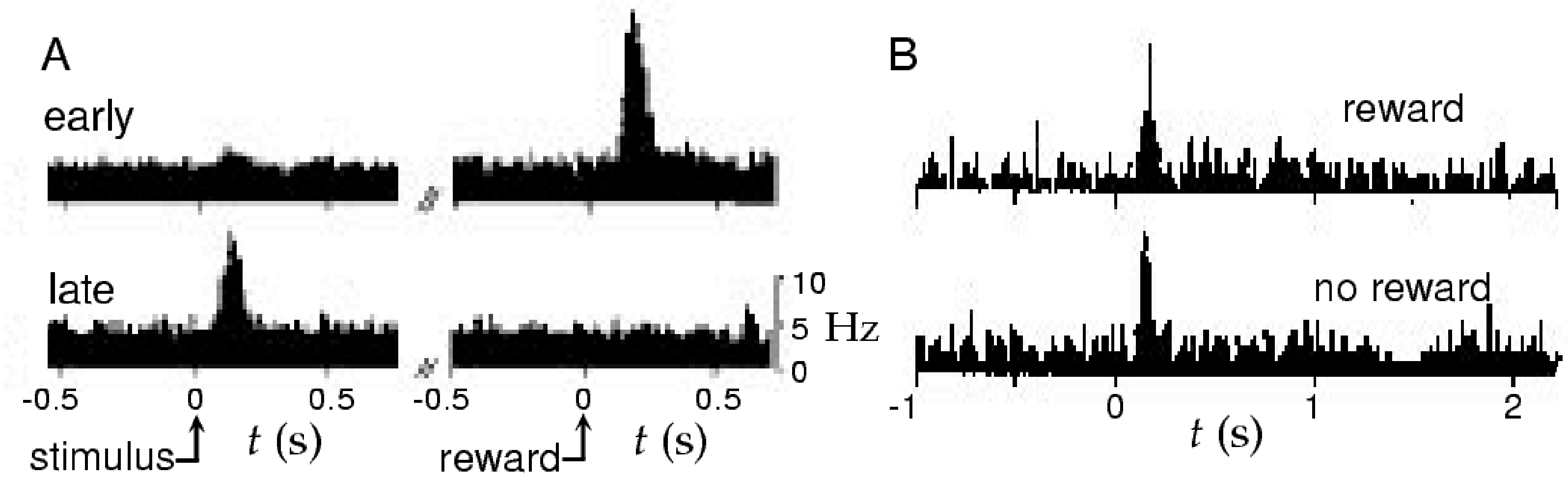
TD

V(B)=3/4

V(A)=3/4?

after Sutton and Barto 1998

Aside: what makes a TD error?



- unpredicted reward expectation change
- disappears with learning
- stays with probabilistic reinforcement
- sequentiality
  - TD error vs prediction error
- see Niv and Schoenbaum 2008

Schultz et al.

TD learning

$$\boxed{a_t \sim \pi(a|s_t)}$$

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

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

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

$$\rightarrow V^{\pi}(s)$$

TD learning

$$\boxed{a_t \sim \pi(a|s_t)}$$

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

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

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

$$\rightarrow V^{\pi}(s)$$

$$\pi^{new}?$$

TD learning

$$\boxed{a_t \sim \pi(a|s_t)}$$

$$s_{t+1} \sim \mathcal{T}^{a_t}_{s_t,s_{t+1}}$$

$$r_t = \mathcal{R}(s_{t+1}, a_t, s_t)$$

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

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

$$\rightarrow V^\pi(s)$$

$\pi^{new}?$

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

SARSA

- Do TD for state-action values instead:

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

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

- base policy on Q

$$p(a|s) = \frac{e^{\beta \mathcal{Q}(s,a)}}{\sum_{a'} e^{\beta \mathcal{Q}(s,a')}} \qquad p(a|s) = \begin{cases} 1-\epsilon & \text{if } a = a^* \\ \epsilon & \text{else} \end{cases}$$

- convergence guarantees

Q learning: off-policy

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

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

Actor-critic

- policy and value separately parametrised

$$\pi(s,a) = \frac{e^{w(s,a)}}{\sum_{a'} e^{w(s,a')}}$$

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$

$$w(s,a) \leftarrow w(s,a) + \beta\delta_t$$

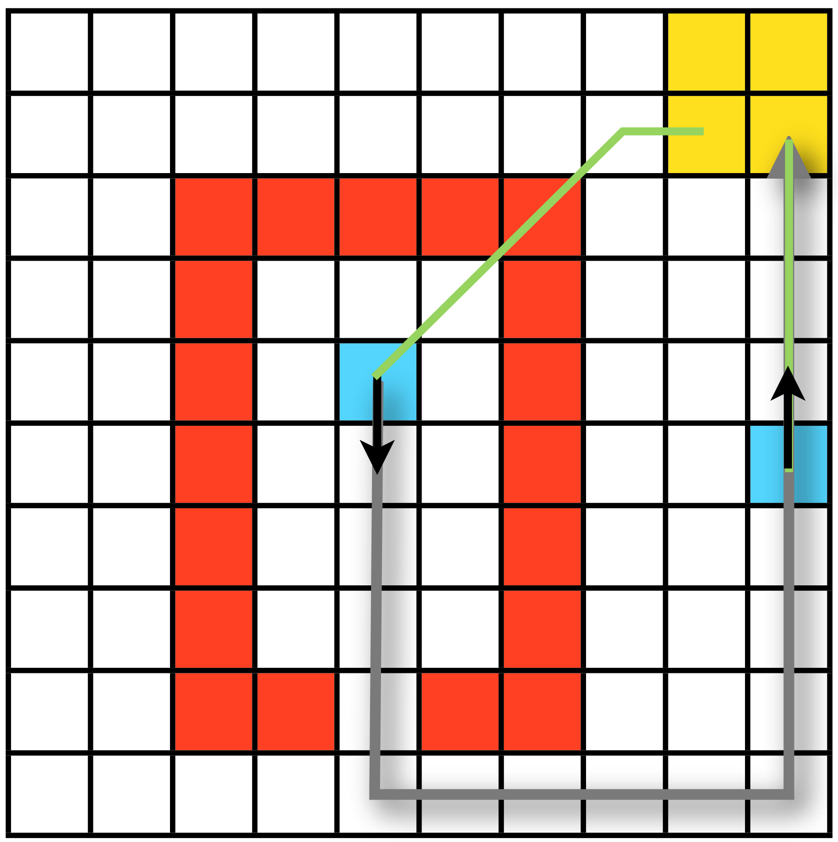
$$w(s,a) \leftarrow w(s,a) + \beta\delta_t(1-\pi(s,a))$$

States

- Some more comments...

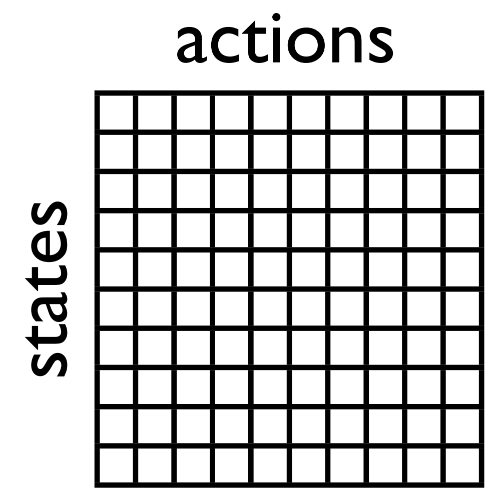
Learning in the wrong state space

- states=distance from goal
- state-space choice crucial
    - too big -> curse of dimensionality
    - too small -> can't express good policies
    - unsolved problem
- humans in tasks have to infer state-space

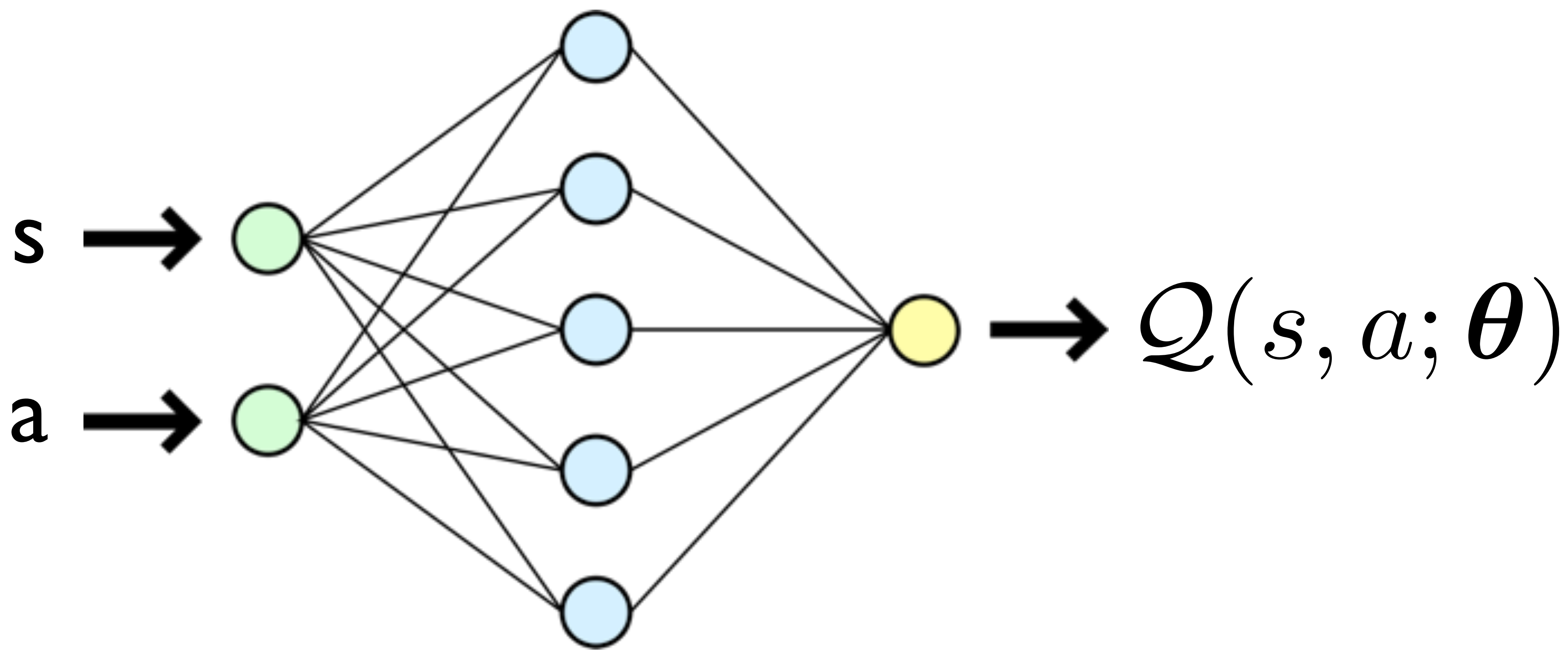


Neural network approximations

- So far: look-up tables



- Parametric value functions



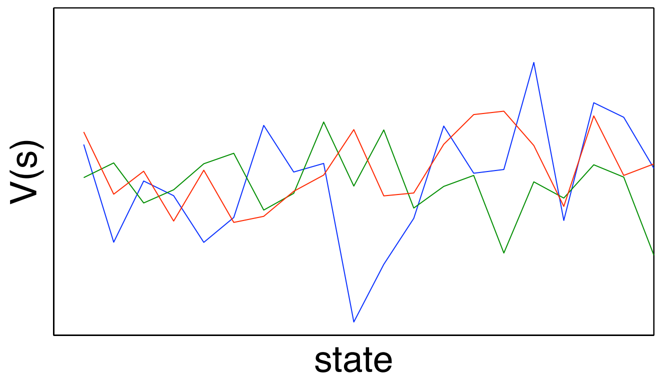
Neural network approximations

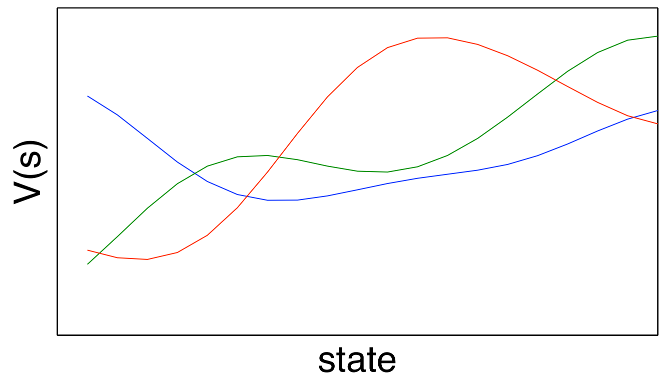
- still get same error: update towards consistent values

$$\delta_t = r_t + V_t(s') - V_t(s_t)$$

- but when doing update, need to apportion responsibility correctly

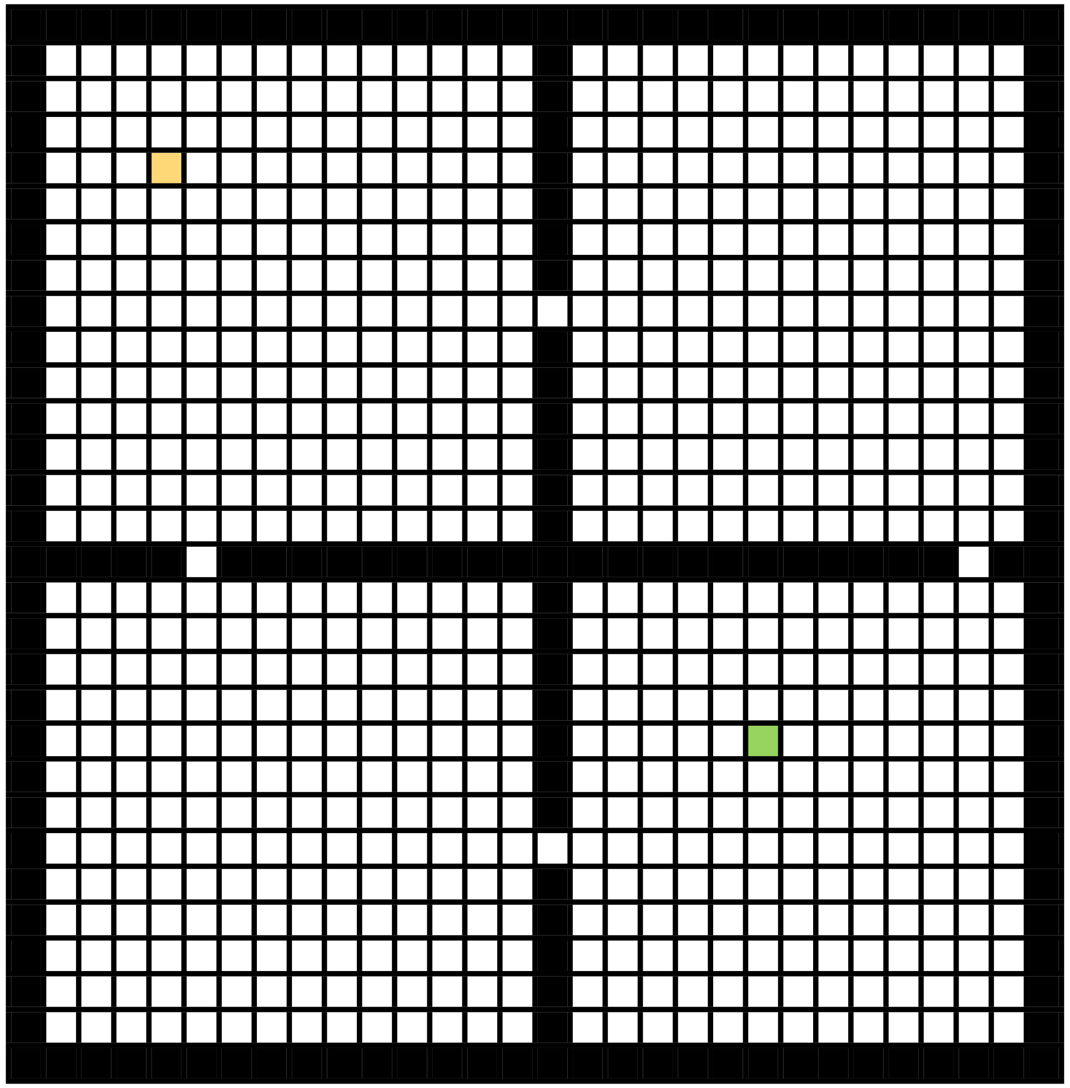
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\delta_t \underbrace{\nabla_{\boldsymbol{\theta}} V_t(s_t)}_{\text{backprop}}$$





Hierarchical decompositions

- Subtasks stay the same
  - Learn subtasks
  - Learn how to use subtasks
- Macroactions
  - ‘go to door’
  - search goal



Learning a model

- So far we've concentrated on model-free learning
- What if we want to build some model of the environment?

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

- Count transitions

$$\hat{\mathcal{T}}^a_{ss'} = \frac{\sum_t \mathbf{1}(s_t = s, a_t = a, s_{t+1} = s')}{\sum_t \mathbf{1}(s_t = s, a_t = a)}$$

- Average rewards

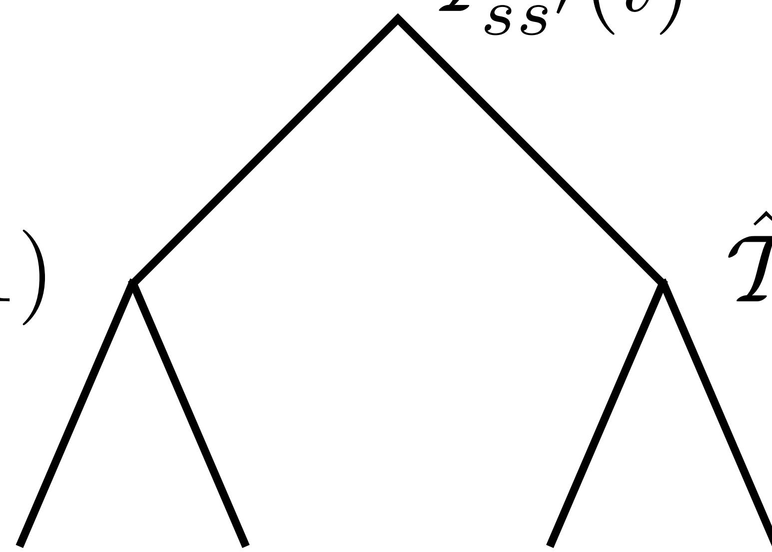
$$\hat{\mathcal{R}}^a_{ss'} = \frac{\sum_t r_t \mathbf{1}(s_t = s, a_t = a, s_{t+1} = s')}{\sum_t \mathbf{1}(s_t = s, a_t = a, s_{t+1} = s')}$$

Using a learned model

- explicitly addresses exploration / exploitation

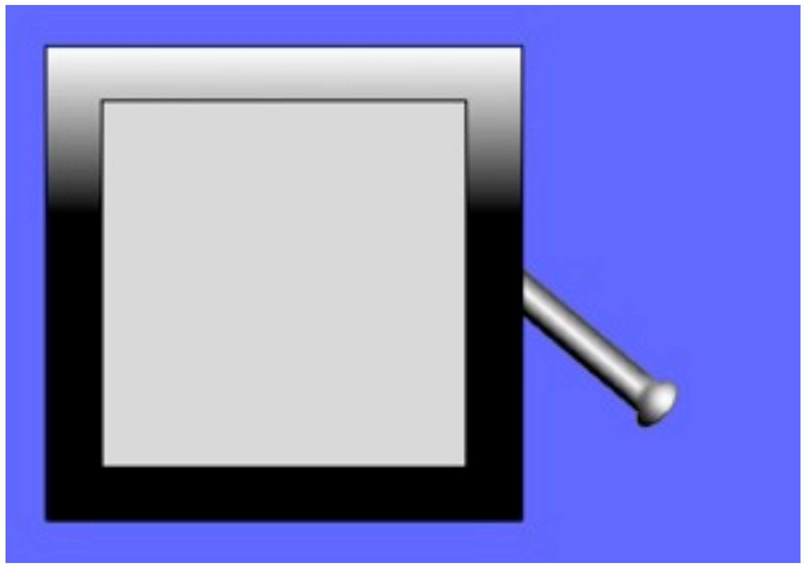
$$\hat{\mathcal{T}}^a_{ss'}(t)$$

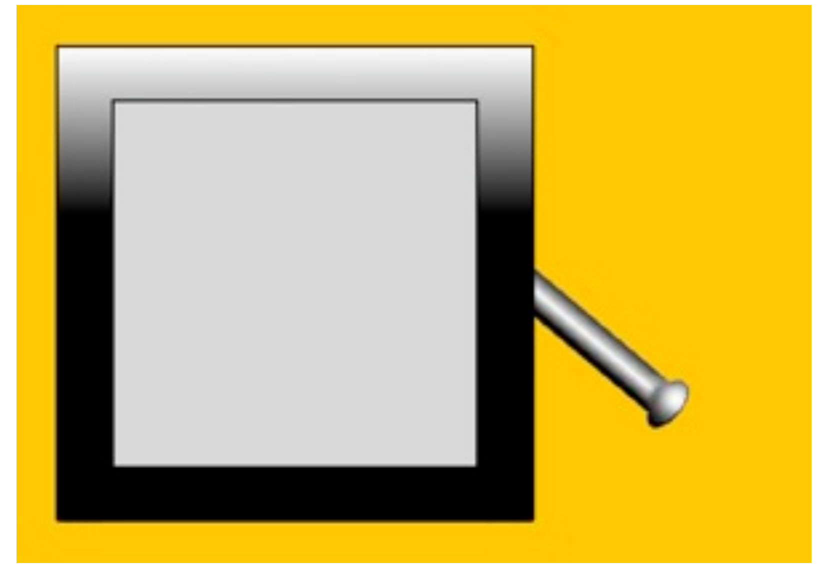
$$\hat{\mathcal{T}}^a_{ss'}(t+1, s'=1) \qquad \hat{\mathcal{T}}^a_{ss'}(t+1, s'=2)$$

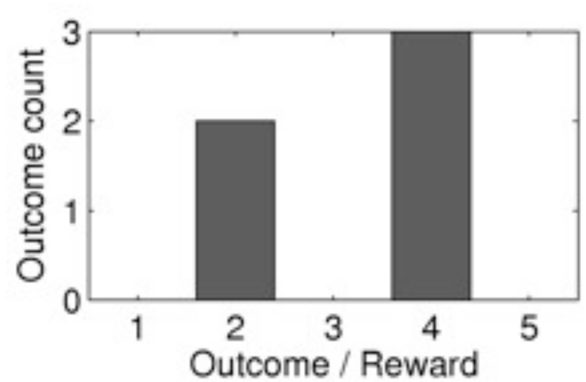


- Model changes as we ‘think ahead’
  - account for the value of added information

Model uncertainty

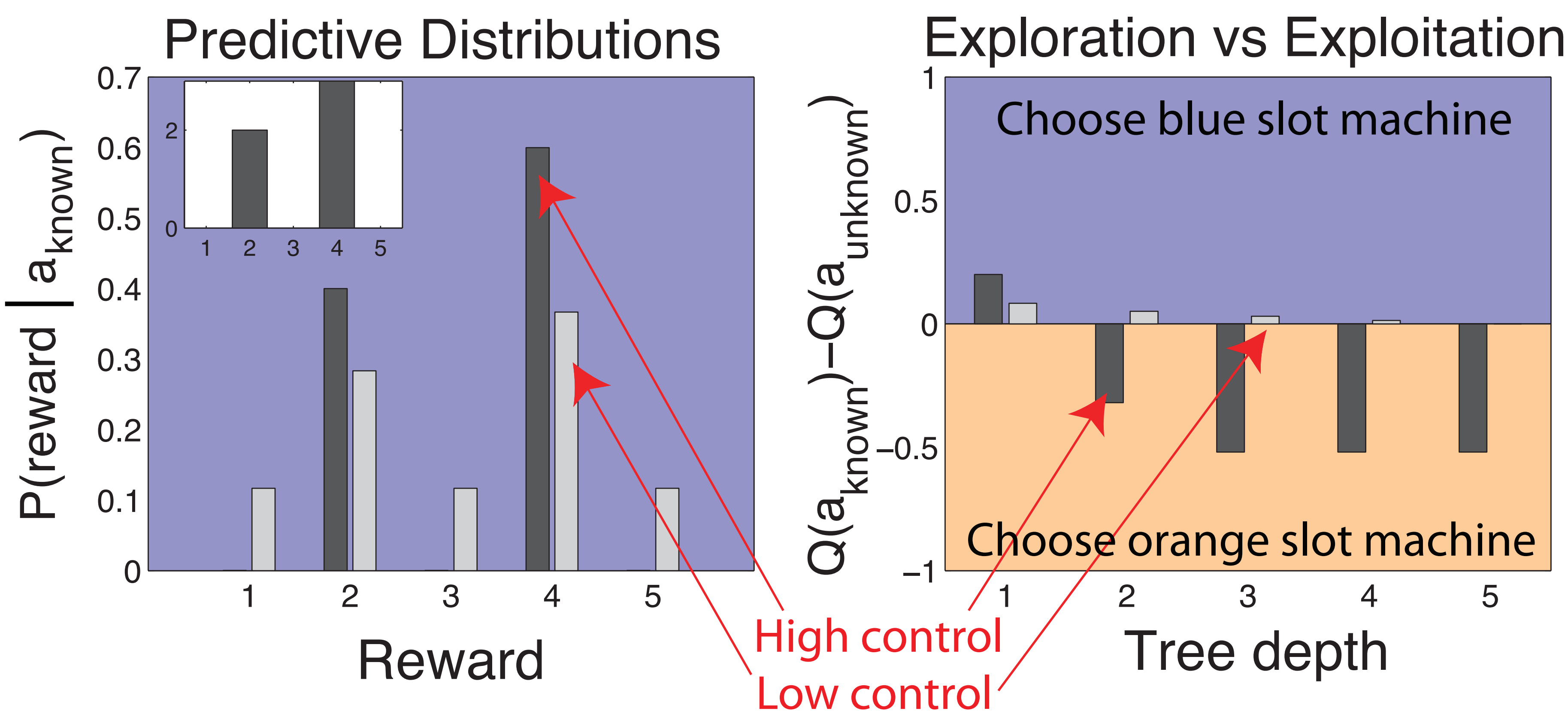






$$\mathcal{Q}(s,a|\hat{\mathcal{T}},\hat{\mathcal{R}}) = \sum_{s'} \hat{\mathcal{T}}^{a}_{ss'}(t)\left[\hat{\mathcal{R}}(s',a,s)(t) + \max_{a'} \mathcal{Q}(s',a'|\hat{\mathcal{T}}(t+1),\hat{\mathcal{R}}(t+1))\right]$$

Consequences of control

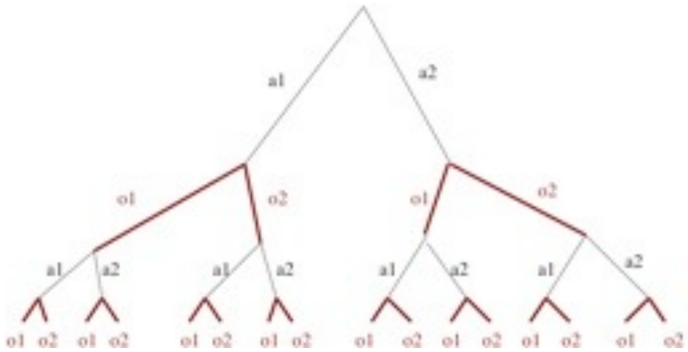


Multiple, parallel, decision-making systems

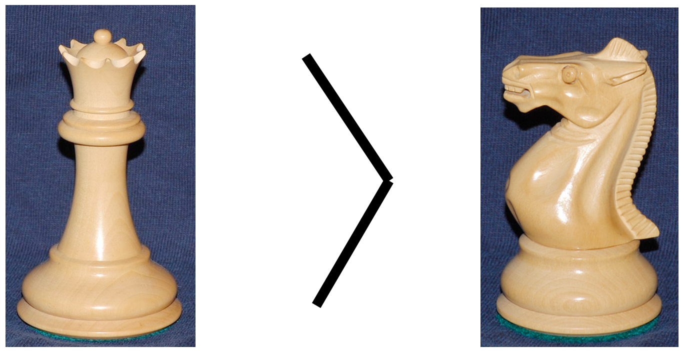
Multiple decision systems "Controllers"

Competition and collaboration

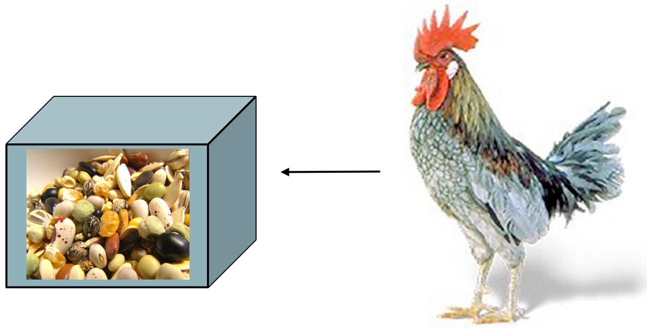
Goal-directed system
Tree search



Habit system
Experience average



Innate system
Evolutionary strategy



In humans, animals and computers...

Some behavioural signatures of different models

Quentin Huys

Wellcome Trust Centre for Neuroimaging
Gatsby Computational Neuroscience Unit
Medical School
UCL

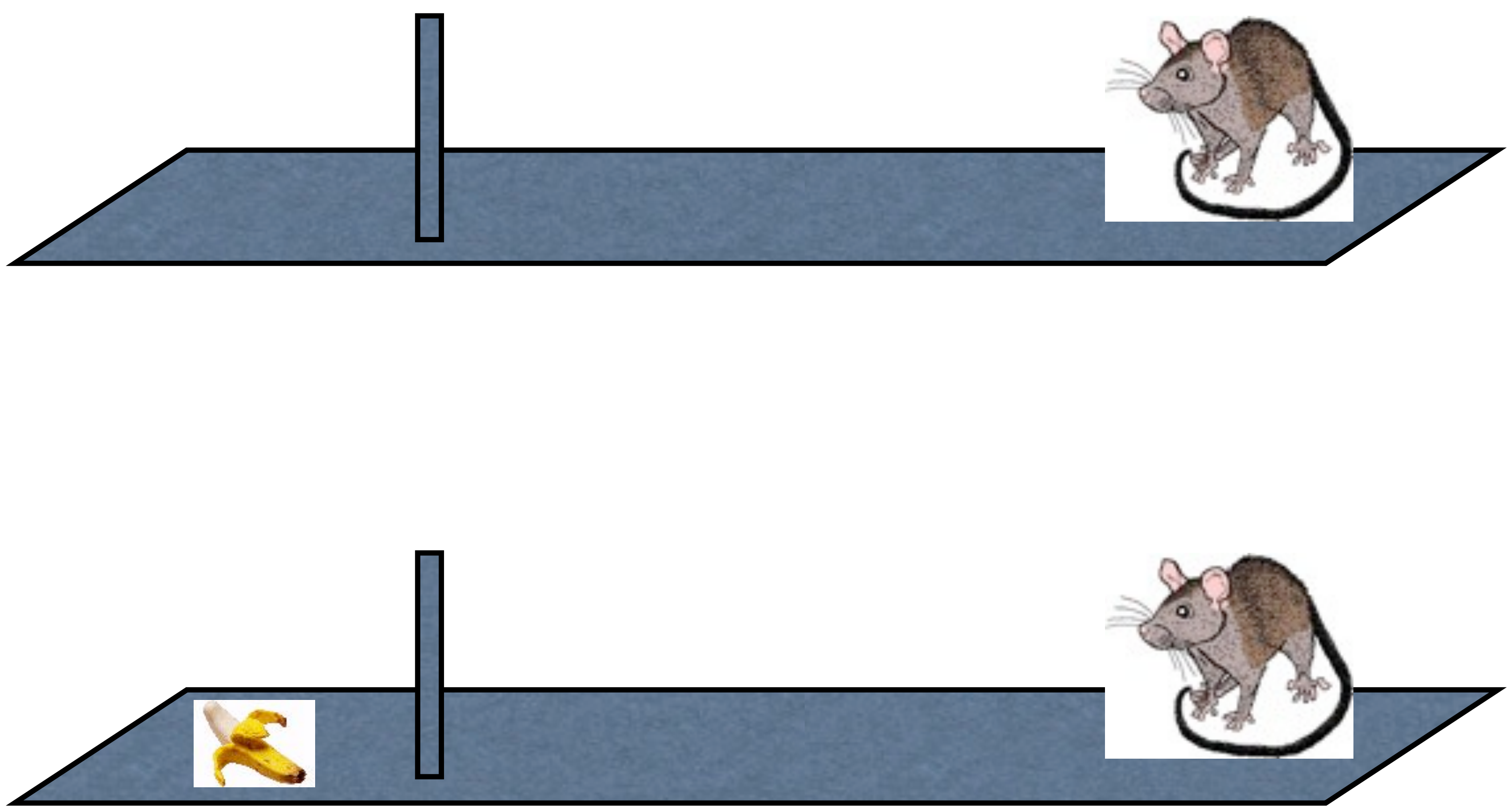
Magdeburg University, June 20th 2009

Why are choices hard?

Time present and time past
Are both perhaps present in time future,
And time future contained in time past.

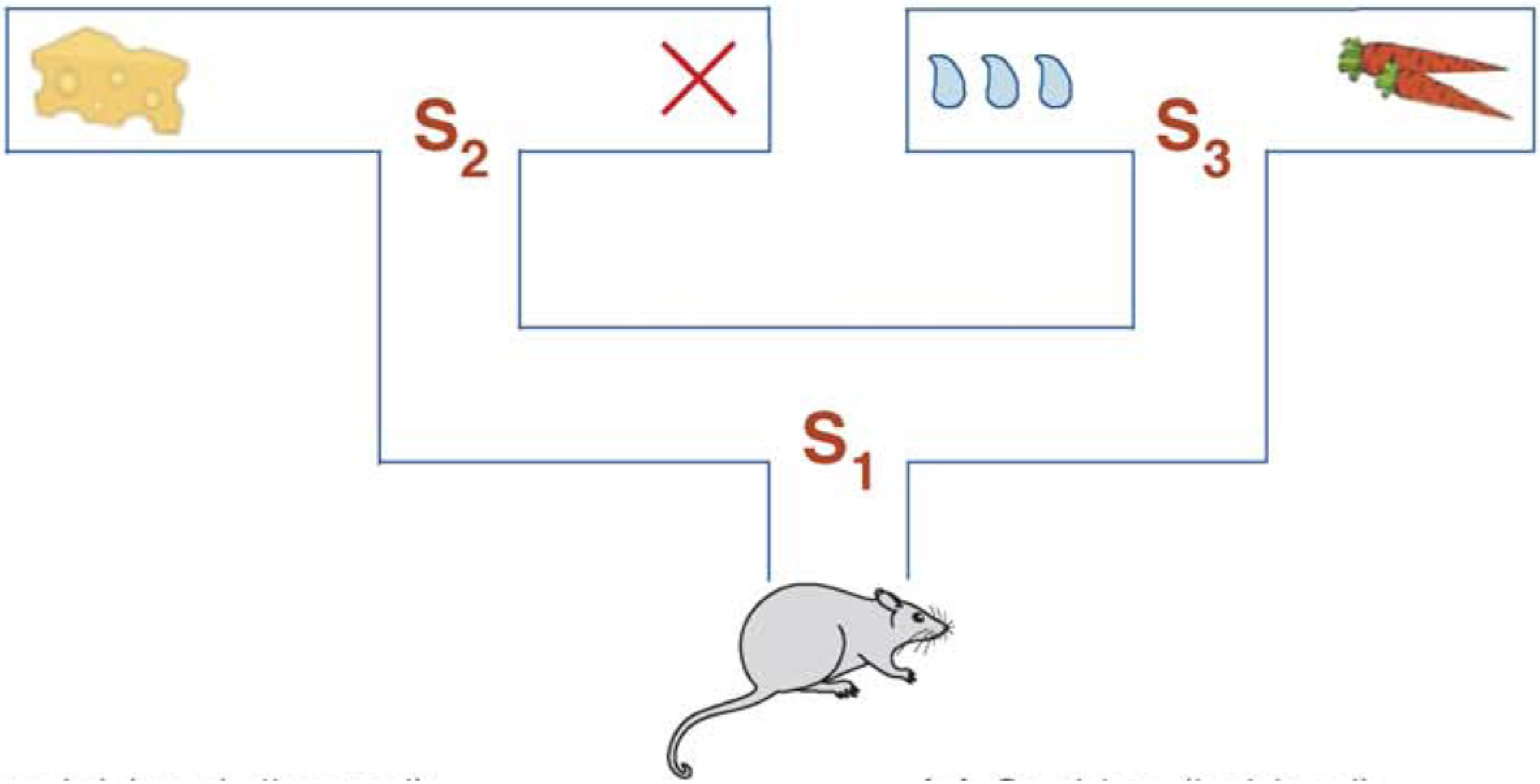
T. S. Eliot

The future, in the long term



goodness of an action = immediate reward + all future reward

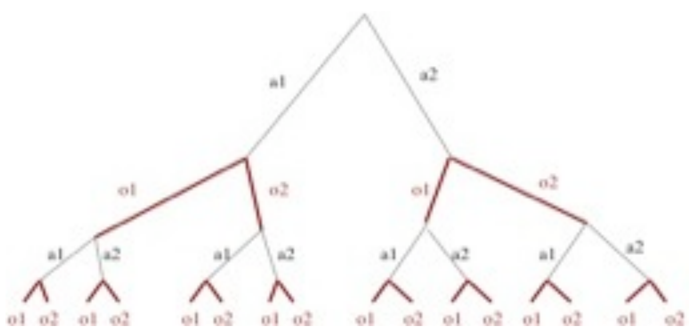
Making optimal decisions



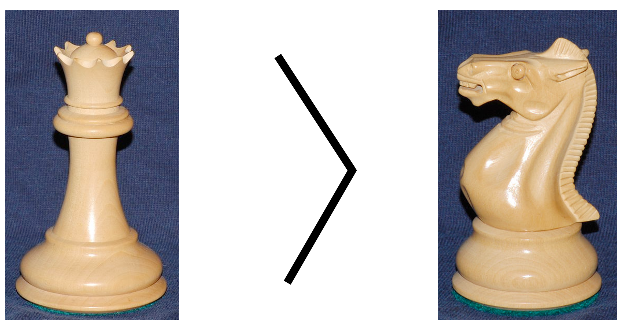
Niv et al. 2007

Many decision systems in parallel

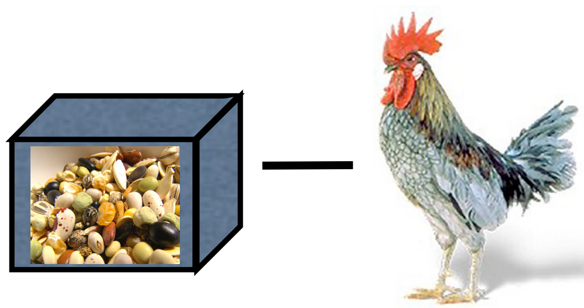
Goal-directed system
Tree search



Habit system
Experience average

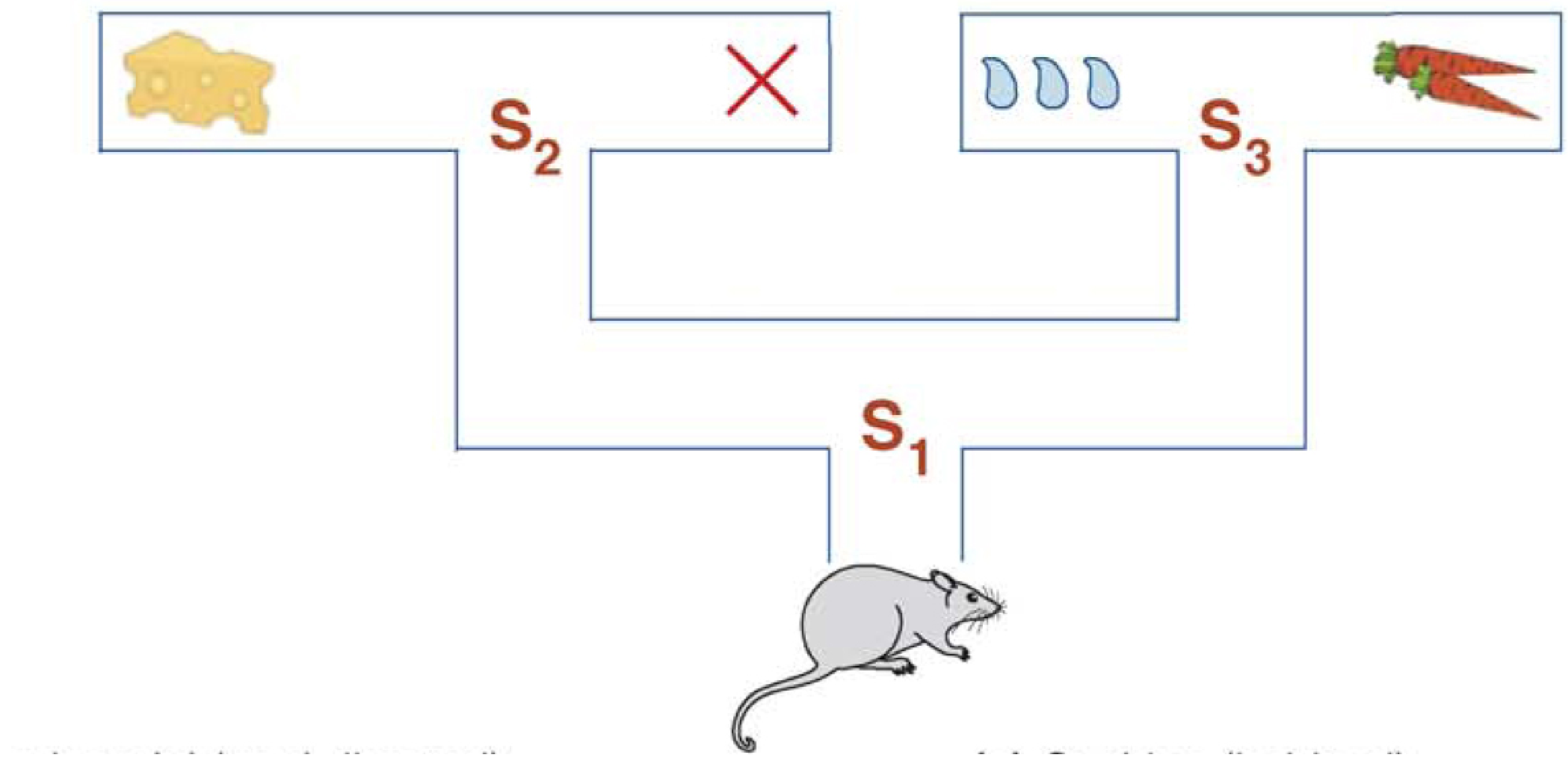


Innate system
Evolutionary strategy



Evaluating the future: Think hard

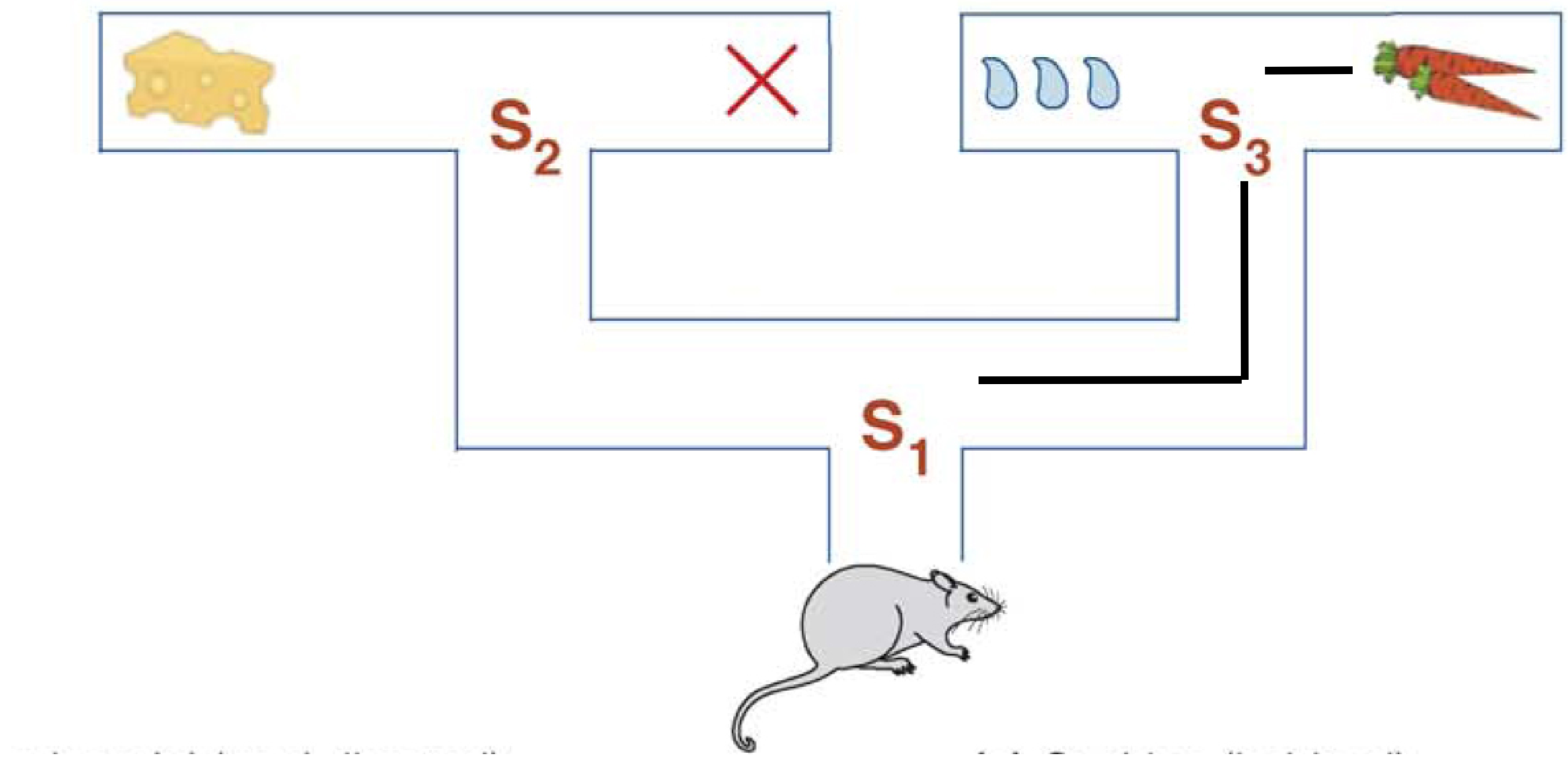
Goal-directed decisions



General solution: search a tree

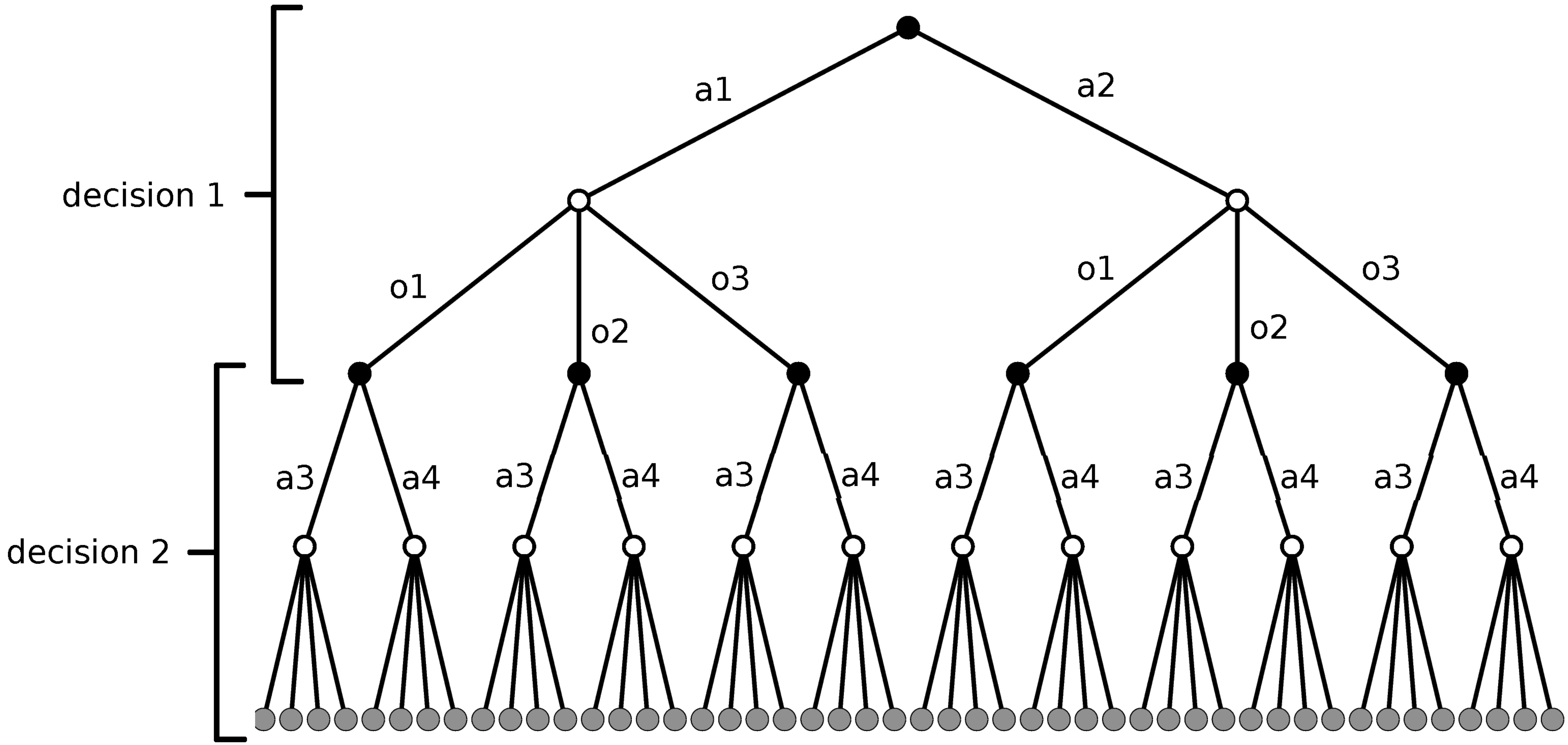
Evaluating the future: Think hard

Goal-directed decisions



General solution: search a tree

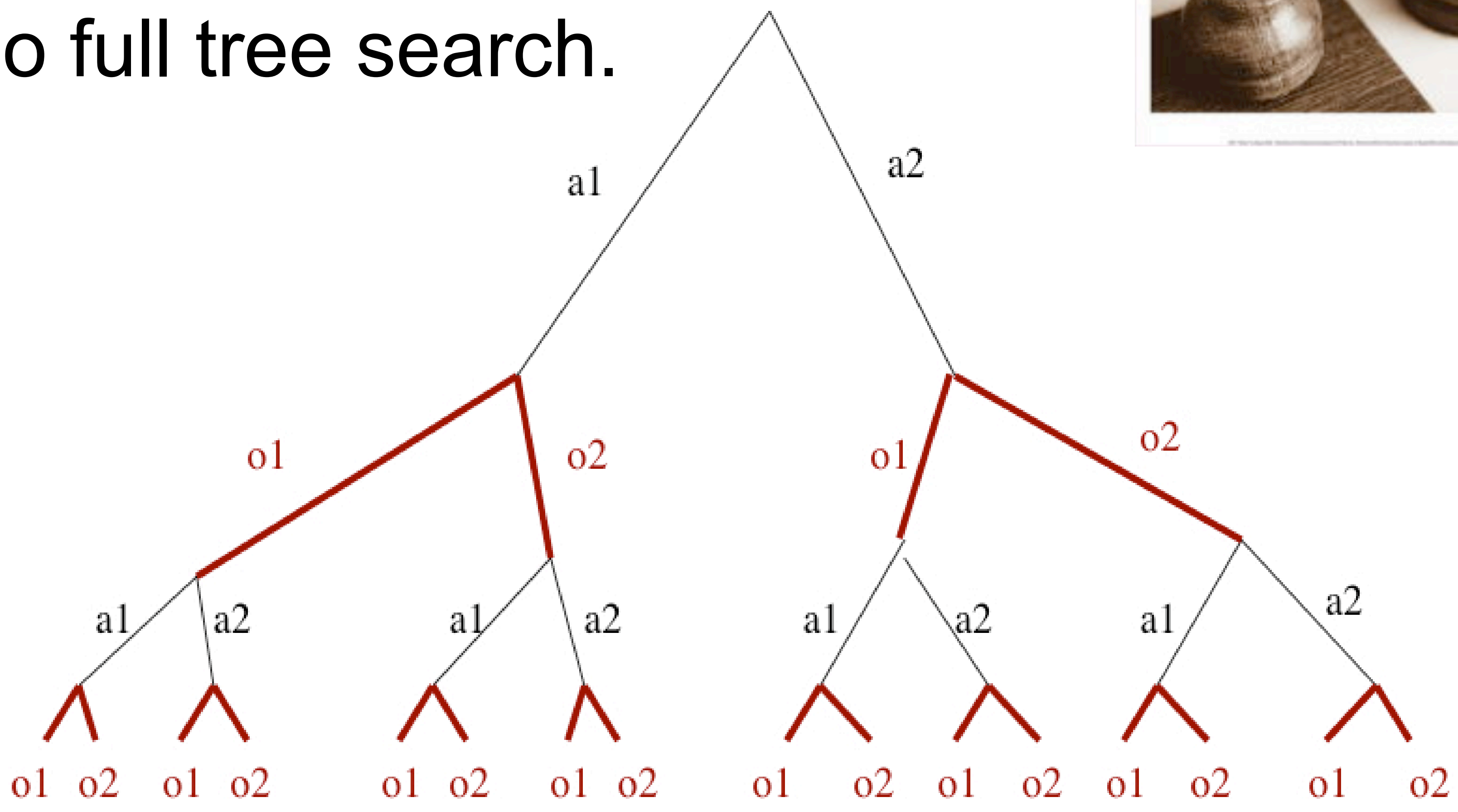
Decision tree: exhaustive search



$$w^d$$

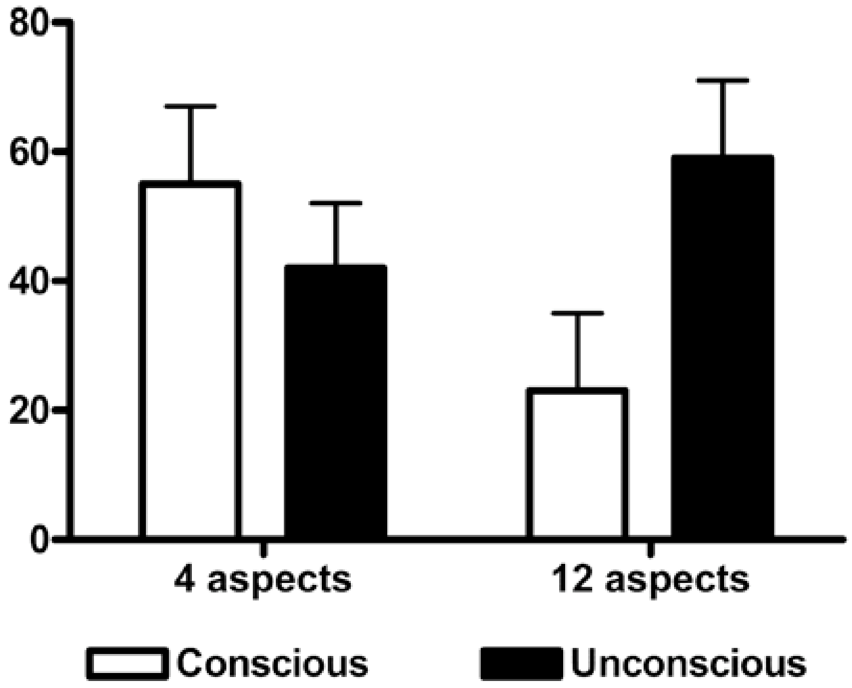
Chess

- Each move 30 odd choices
- 30^{40}?
- MANY!!!
  - Legal boards ~10^{123}
- Can't just do full tree search.



Simple is better at times: cars

Car A: 75% +ve
Car B: 50% +ve
Car C: 50% +ve
Car D: 25% +ve



Asian disease: time

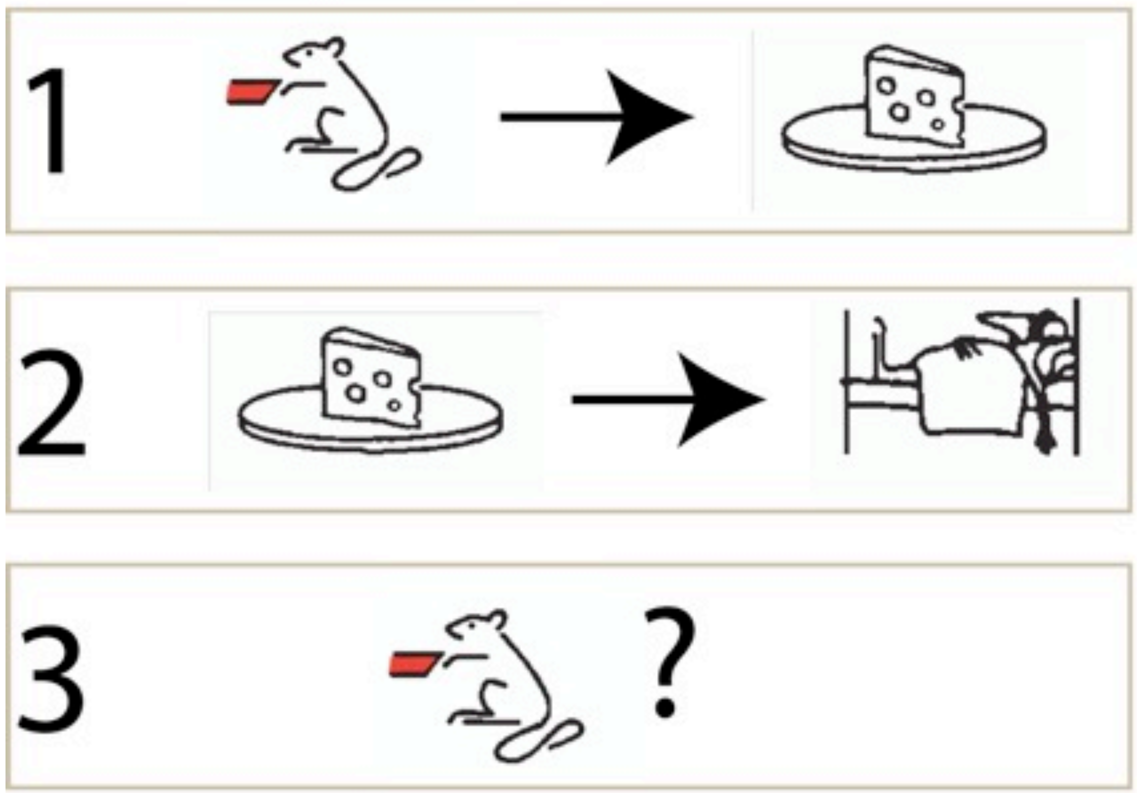
Dijksterhuis et al. 2006

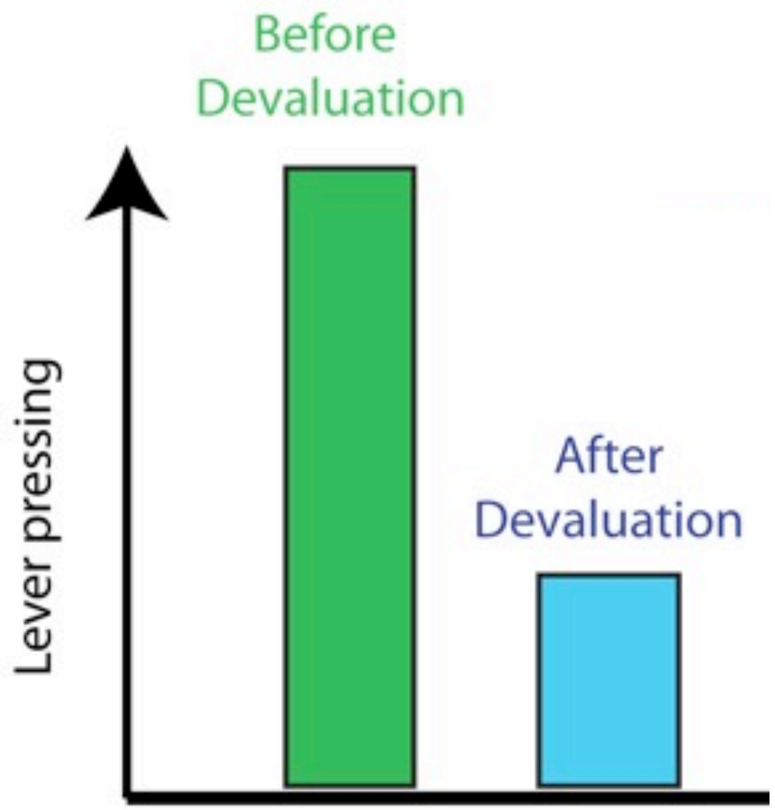
So…?

So...?

How do HUMAN players do it?
How did Deep Blue beat Kasparov?

Devaluation

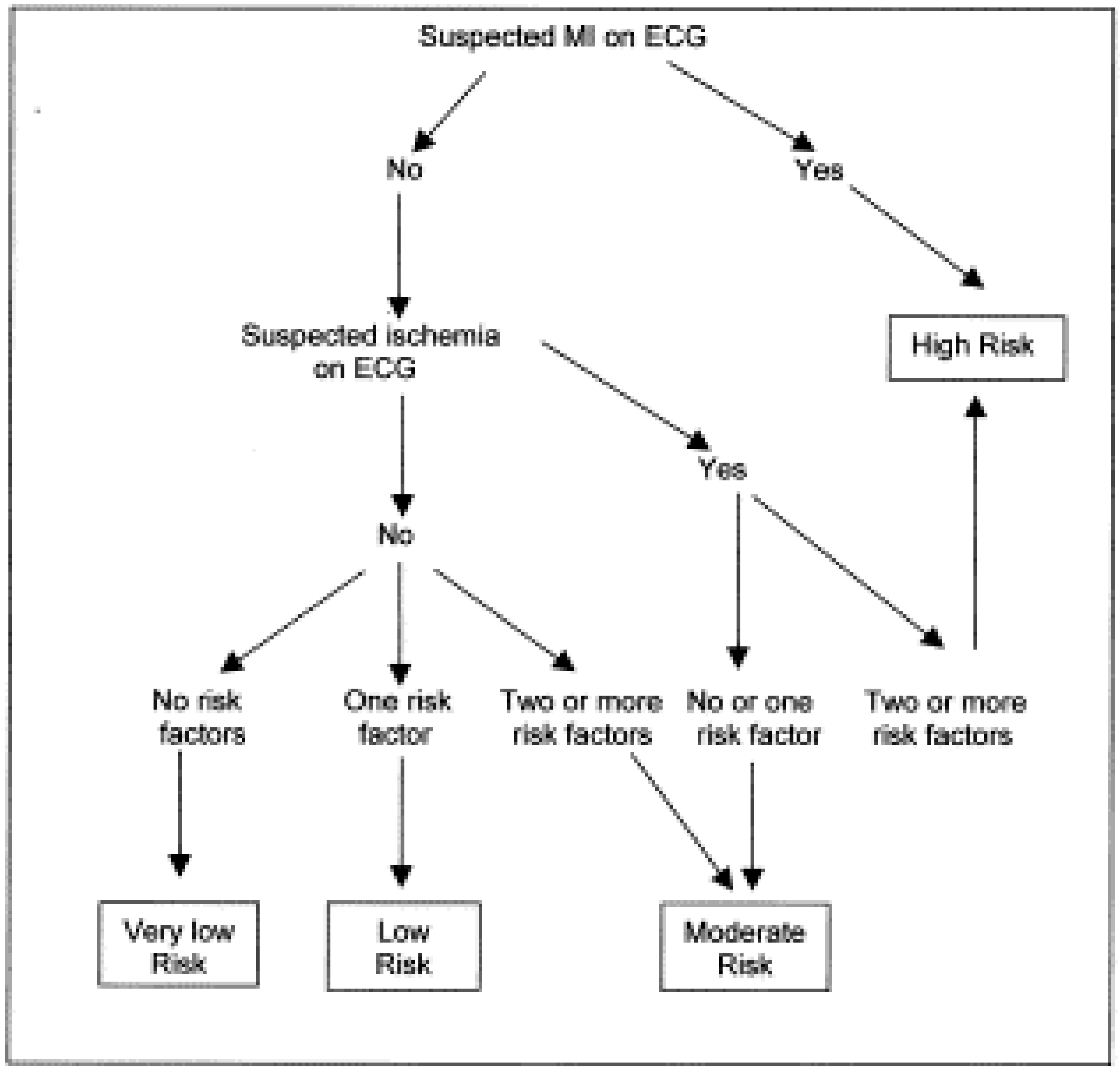




Goal-directed choices

- Model-based
  - how is the model learned?
- Computationally expensive
- Flexible
- Action-outcome

Simple is better at times: doctors

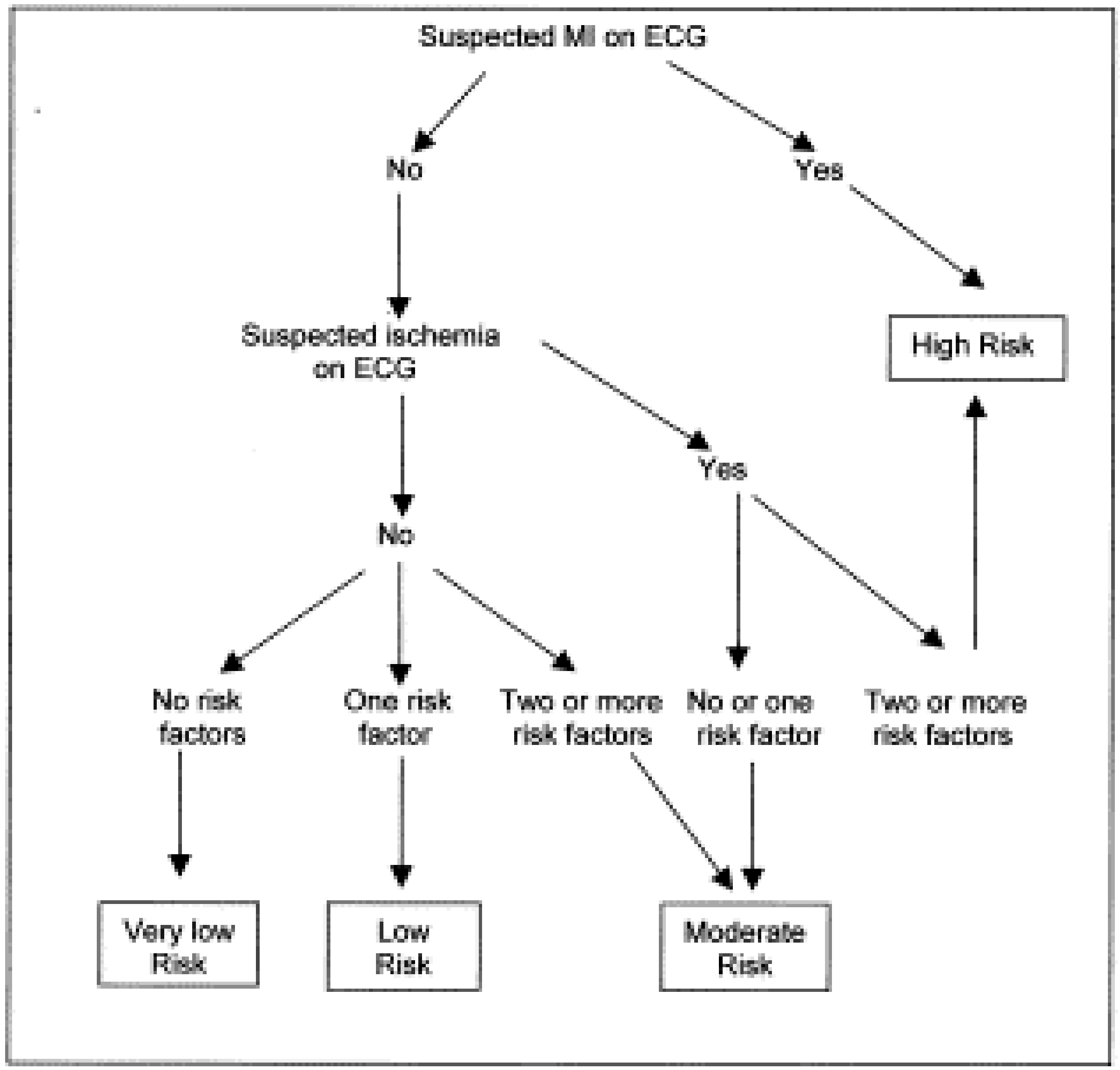


20 cases for which truth known

Cardiologists
General physicians
A&E physicians

Melly et al. 2002

Simple is better at times: doctors



20 cases for which truth known

Cardiologists
General physicians
A&E physicians

Physicians overly cautious, but still miss many -> complications

Melly et al. 2002

Cached evaluation: TD & Co

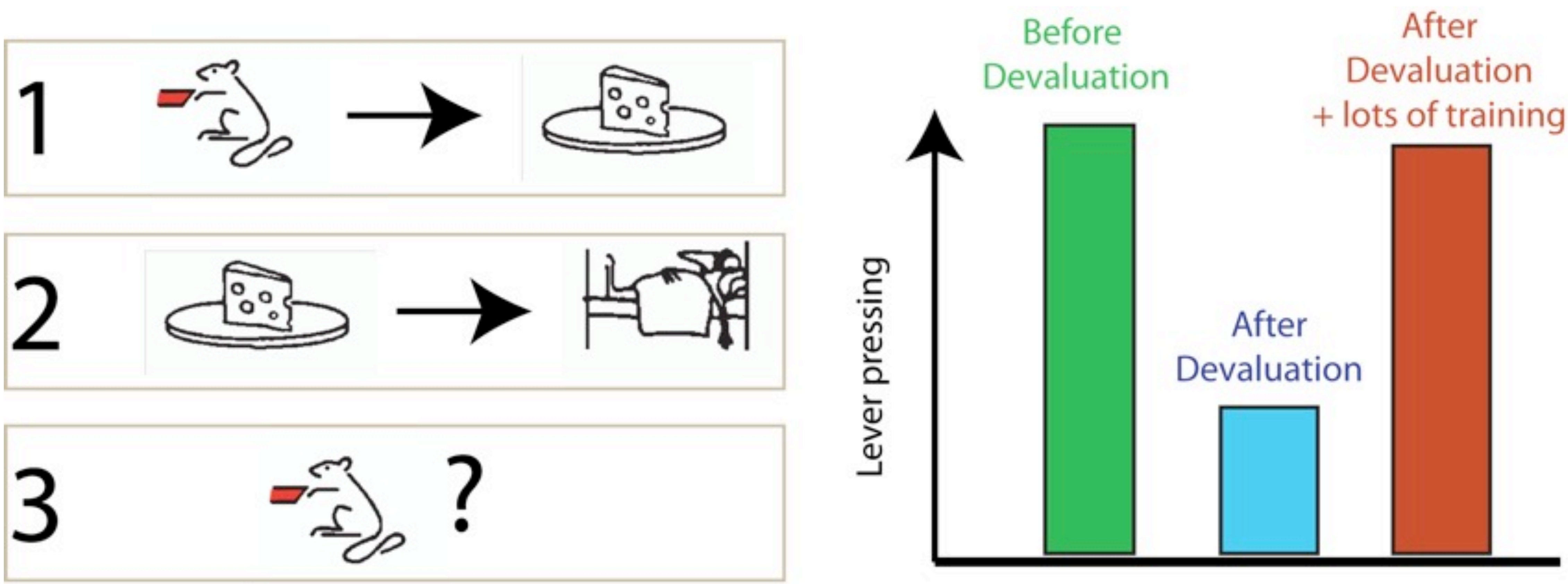
$$\begin{aligned} a_t &\sim \pi(a|s_t) \\ s_{t+1} &\sim \mathcal{T}^{a_t}_{s_t,s_{t+1}} \\ r_t &= \mathcal{R}(s_{t+1}, a_t, s_t) \end{aligned}$$

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

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

Habits: heuristics, position evaluation

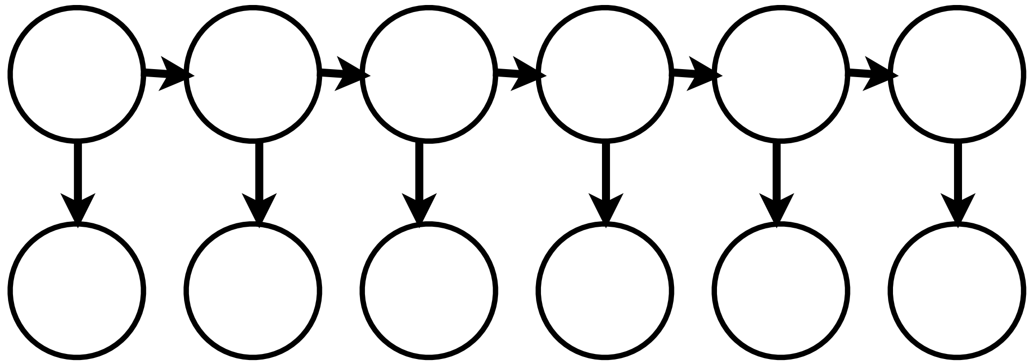
Devaluation



Goal-directed vs. habitual behaviour
mix and match

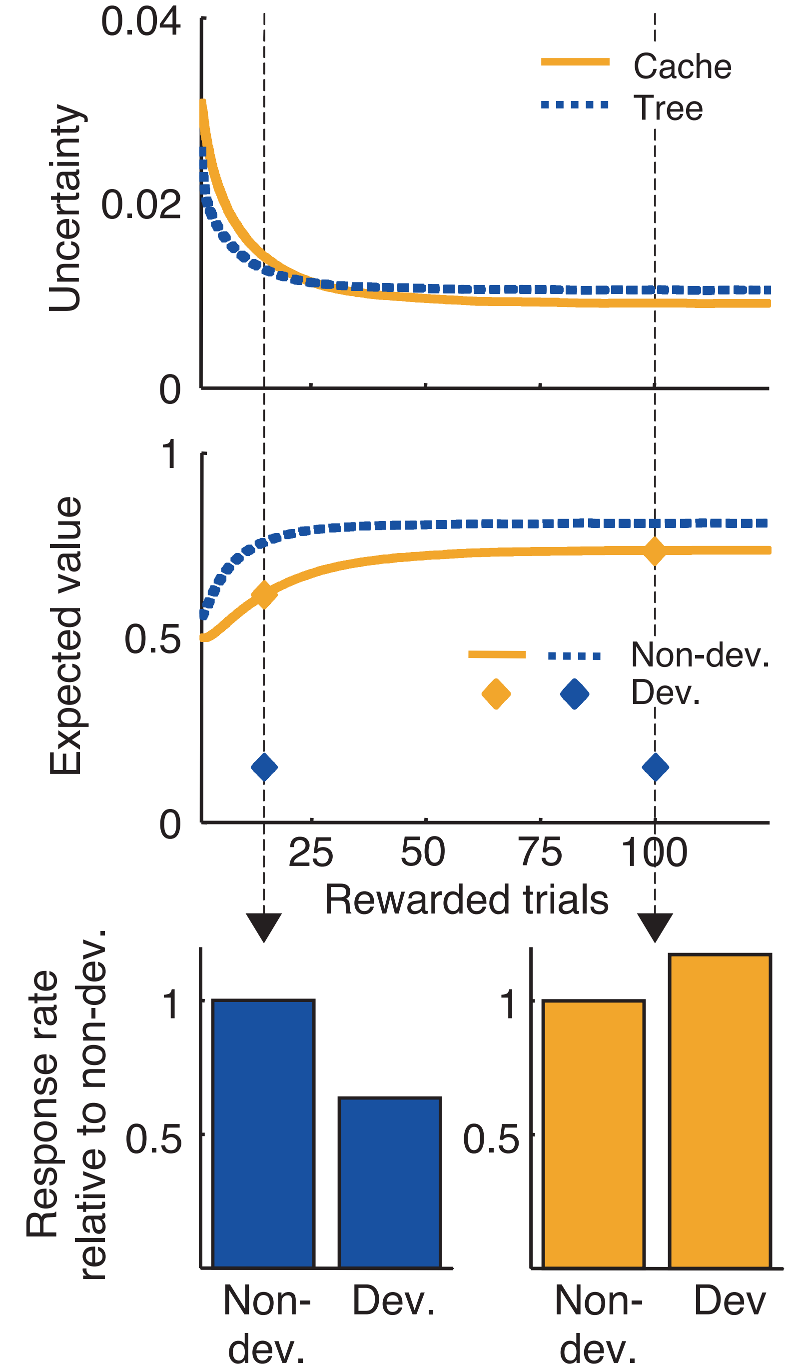
Habits

- Are empirical averages
- Change slowly
- Are cheap to build
- No unlearning
  - extinction
  - higher-order models



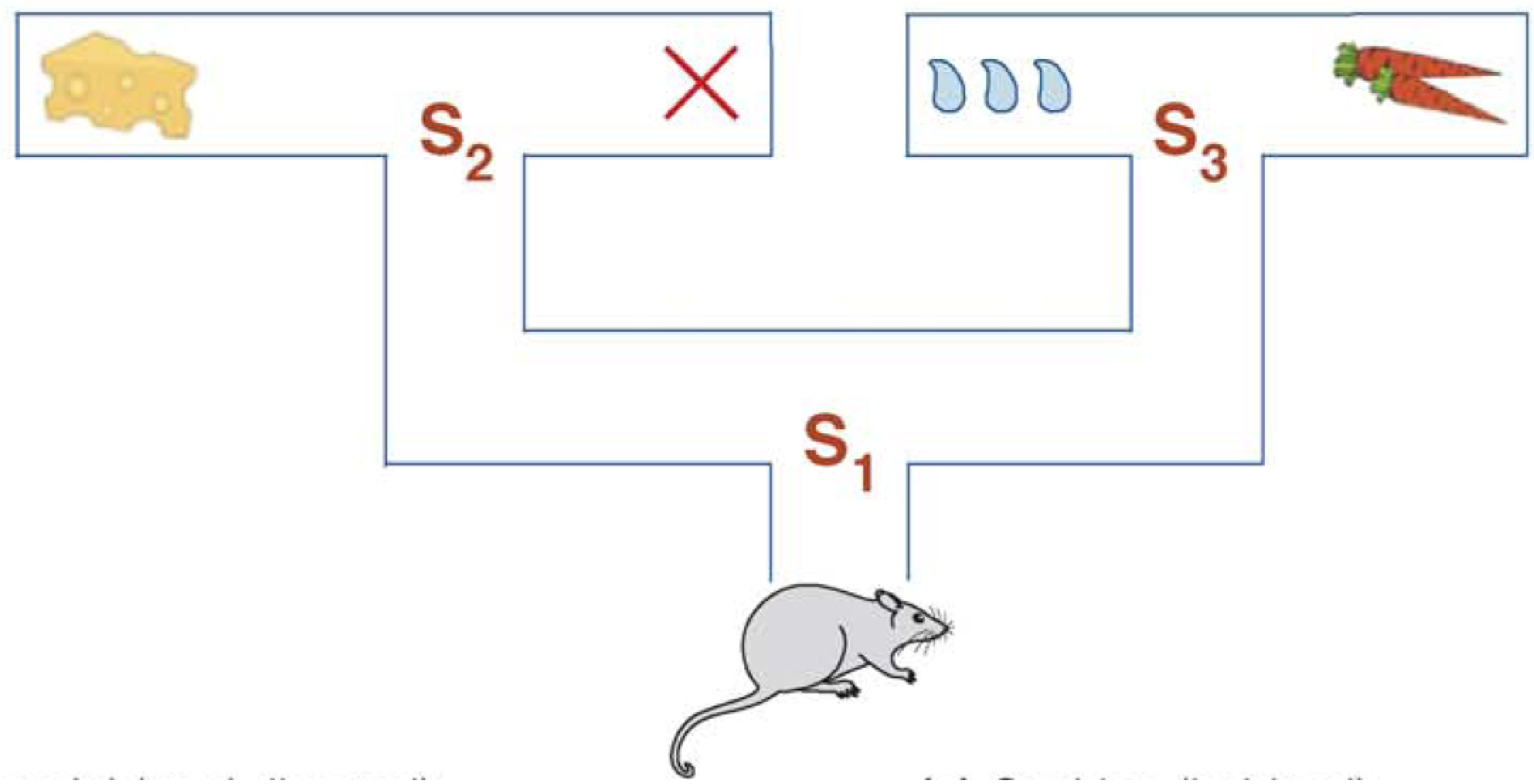
Arbitrating between controllers

- Uncertainty



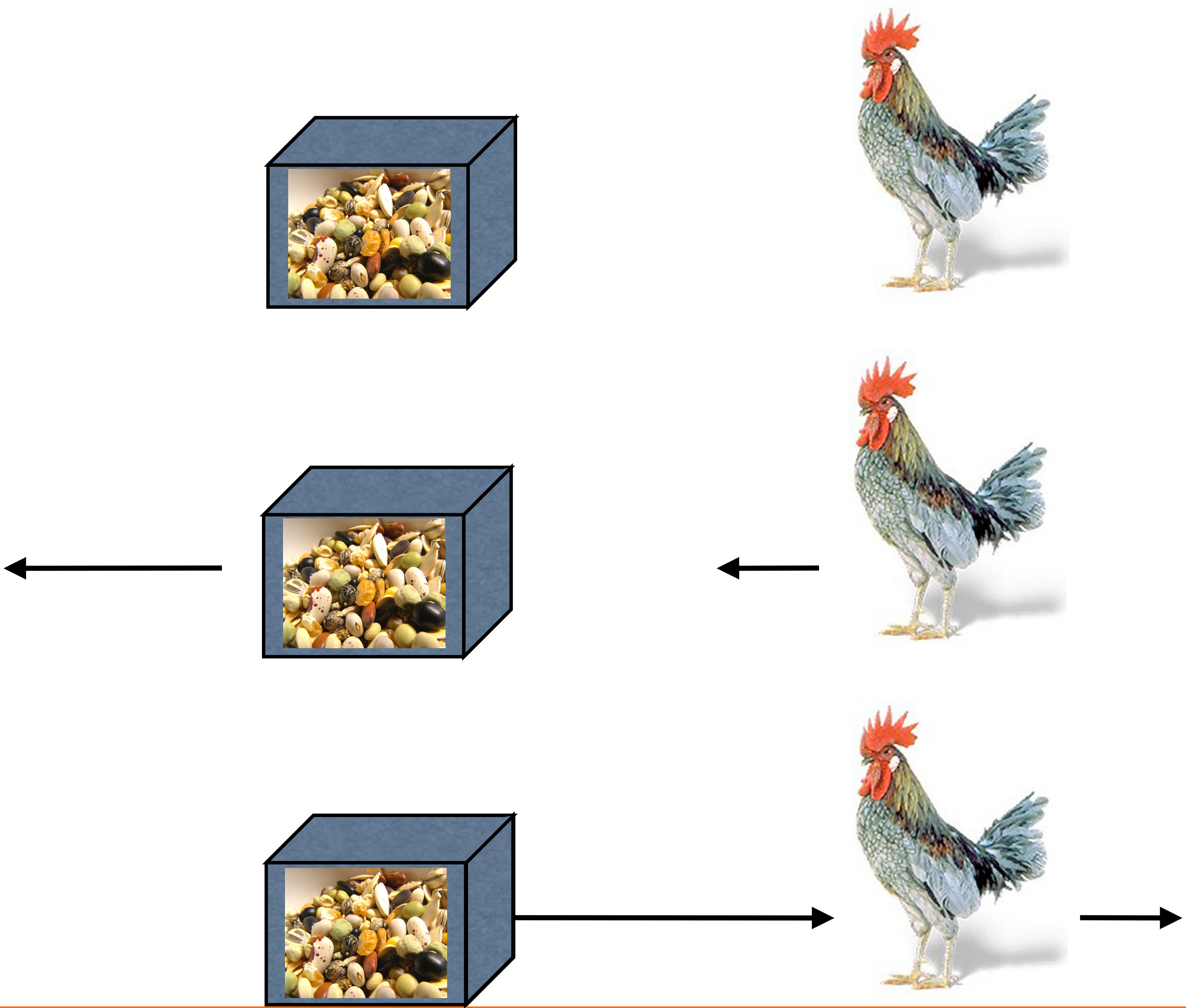
Daw et al. 2005

Evaluating the future… actually, let's not!



Choose randomly at S1
Then just go for food if hungry
Or for water if thirsty

Are chicken pretty stupid?



Hershberger 1986

Kahnemann & Tversky

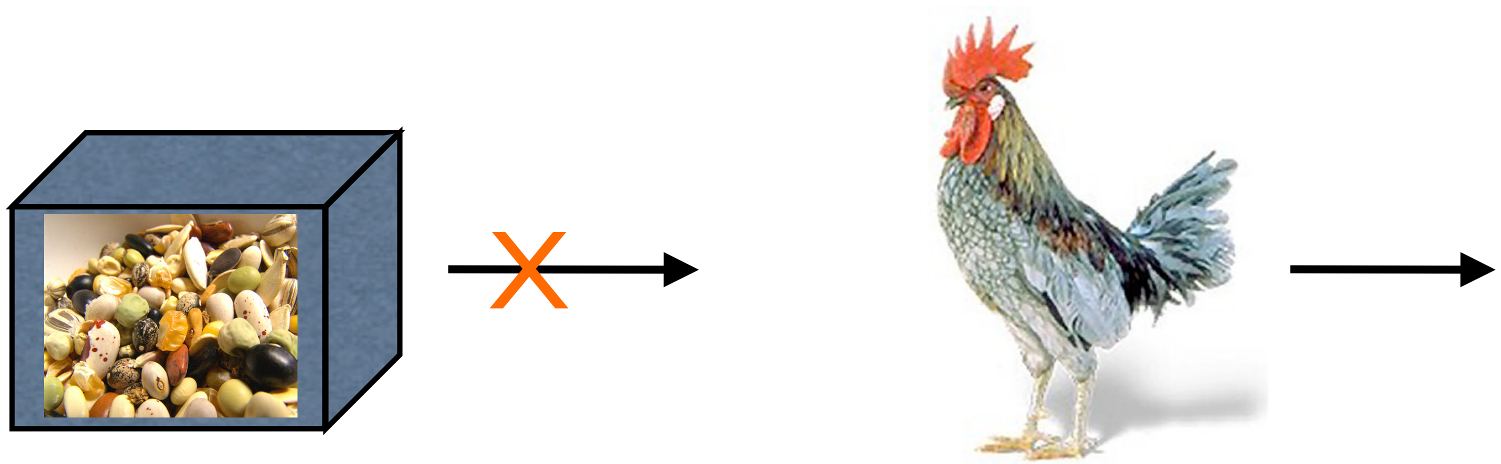
Imagine that the United States is preparing for the outbreak of an unusual Asian disease, which is expected to kill 600 people.

Two alternative programs to combat the disease have been proposed.

Assume that the exact scientific estimates of the consequences of the programs are as follows:

If Program A is adopted, 200 people will be saved If Program B is adopted, there is a one-third probability that 600 people will be saved and a two-thirds probability that no people will be saved.	A
If Program A' is adopted, 400 people will die If Program B' is adopted, there is a one-third probability that nobody will die and a two-thirds probability that 600 people will die	B’

Clever innate strategies



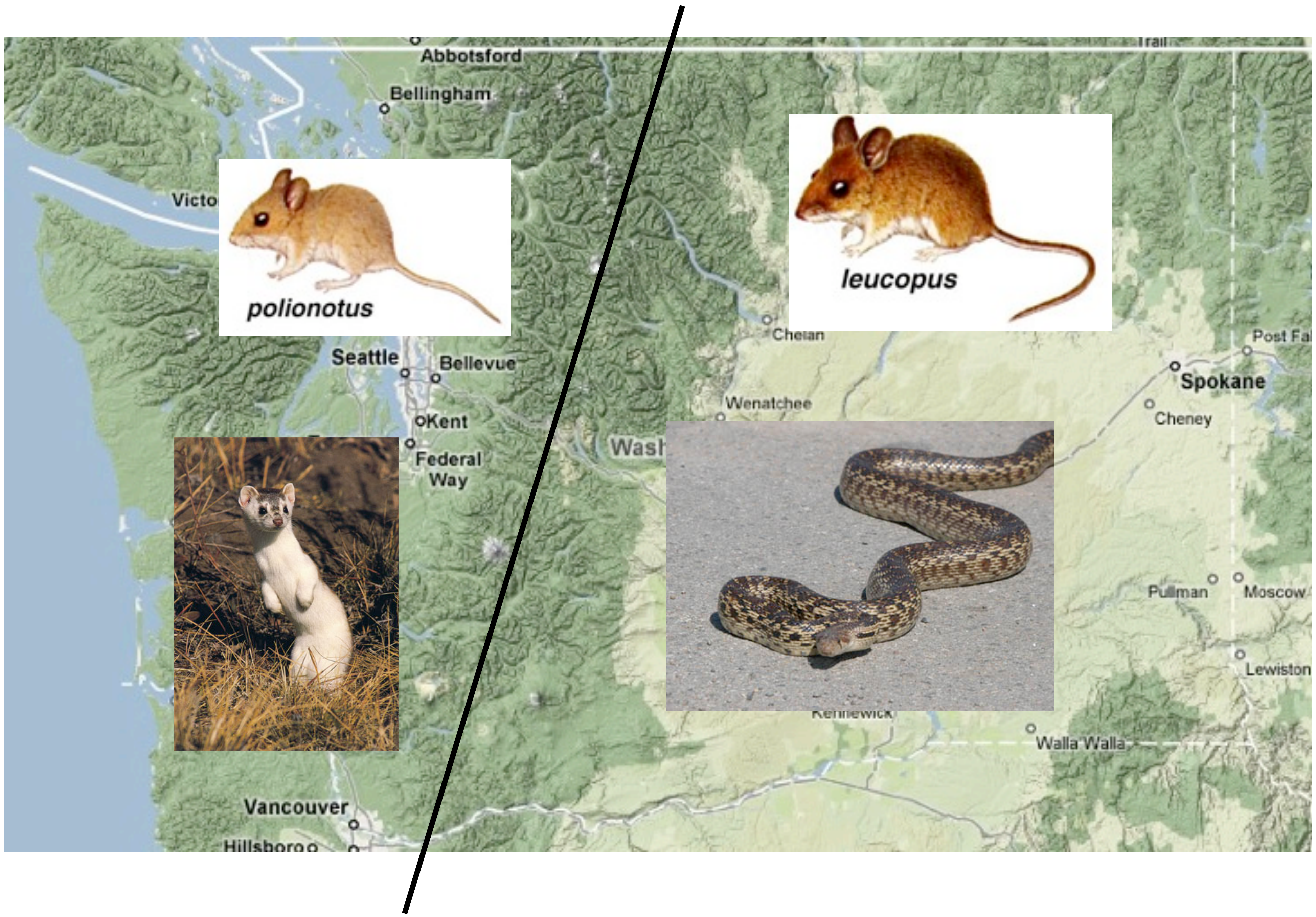
If Program A is adopted, 200 people will be saved
If Program B is adopted, there is a one-third probability that 600 people will be saved and a two-thirds probability that no people will be saved.

A

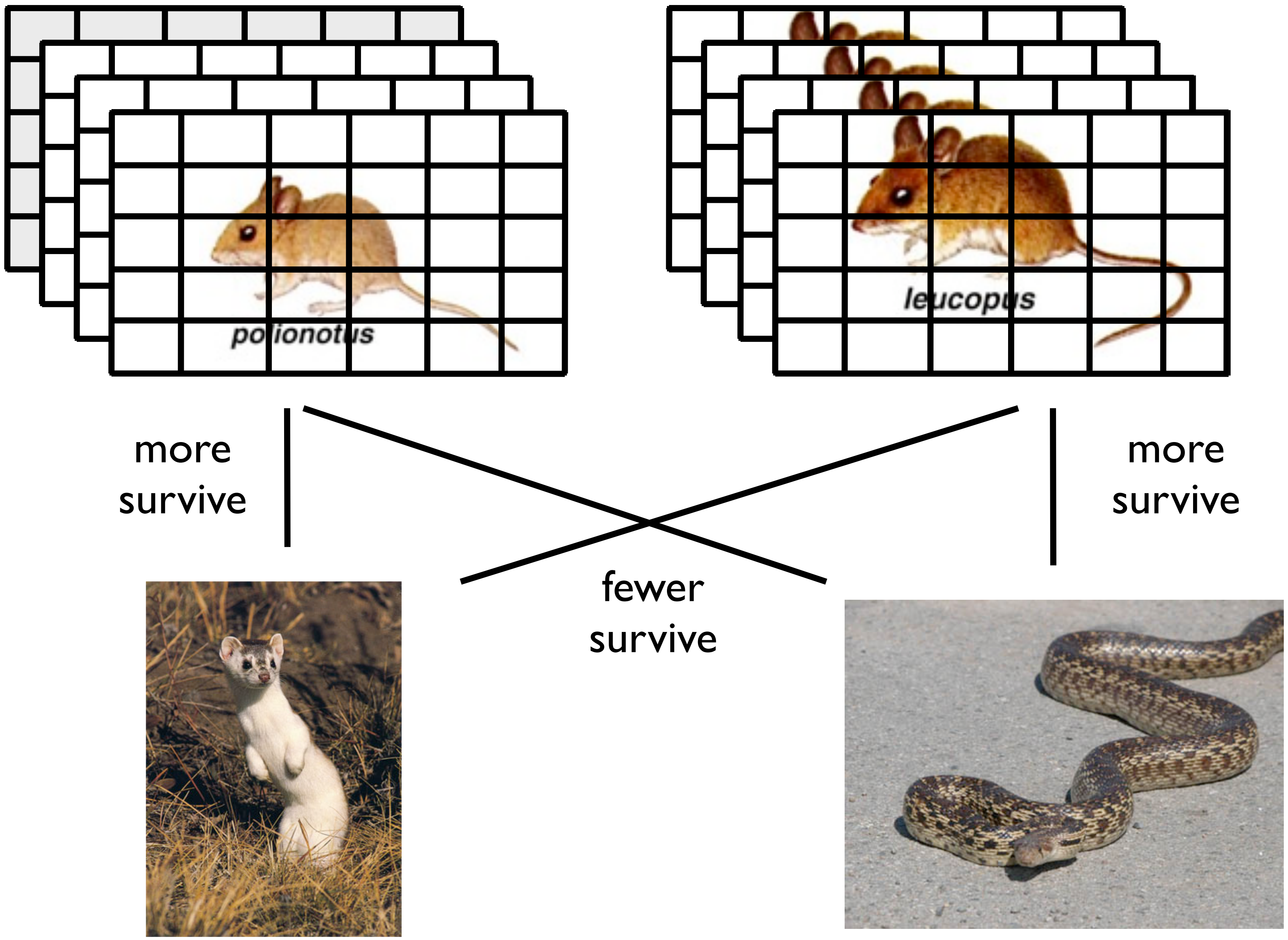
If Program A' is adopted, 400 people will die
If Program B' is adopted, there is a one-third probability that nobody will die and a two-thirds probability that 600 people will die

B’

Innate evolutionary strategies



Innate evolutionary strategies



Hirsch and Bolles 1980

Sometimes knowledge hurts

"We added balsamic vinegar to one of these"

Sometimes knowledge hurts

"We added balsamic vinegar to one of these"

Sometimes knowledge hurts

"We added balsamic vinegar to one of these"

"We added balsamic vinegar to the light one"

Sometimes knowledge hurts

"We added balsamic vinegar to one of these"

"We added balsamic vinegar to the light one"

Recap

- Multiple decision systems
- Multiple values
- Multiple action mechanisms
- Interactions
  - Override
  - Uncertainty
- Complex problem
- Identification via critical features

Fitting behavioural data with RL models

Quentin Huys

Wellcome Trust Centre for Neuroimaging
Gatsby Computational Neuroscience Unit
Medical School
UCL

Magdeburg University, June 20th 2009

Overview

- Formulate probabilistic model for choices
  - model fit: predictive probability
- ML / MAP
  - parameter inference
  - prior inferred from all joint data
- Empirical prior
  - Infer with approximate EM
  - second level analysis:
    - priors
    - individual posterior parameters
- Model comparison
  - Normal-inverse Gamma -> Gaussian mixture

RL models

- 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
  - includes all consequences of a hypothesis for choice

Fitting models: matching and noise

- probabilistic policy, e.g. softmax

$$p(a|s) = \frac{e^{\beta Q(s,a)}}{\sum_{a'} e^{\beta Q(s,a')}}$$

- total likelihood

$$\mathcal{L}(\theta) = p(\{a_t\}_{t=1}^T | \{s_t\}_{t=1}^T, \{r_t\}_{t=1}^T, \theta) = \prod_{t=1}^T p(a_t | s_t, r_{1\dots t-1}, \theta)$$

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \mathcal{L}(\theta)$$

Typical parameters

- r / β

$$
\begin{aligned}
\mathcal{Q}_{t+1}(s,a) &\propto \sum (1-\alpha)^{t-t'} r_{t'} = \eta \sum (1-\alpha)^{t-t'} r'_{t'} \\
r' &= \frac{r}{\eta}
\end{aligned}
$$

  - similar if want to infer $r^+ > 0$ and $r^- < 0$ and separately
  - can only distinguish these with some neural signature
- learning rate α
  - multiplies TD error
  - also induces forgetting
- discounting γ
  - only if there is actually a sequential aspect
- Instructions
- TD error:
  - affected by both r and α

Overview

- Formulate probabilistic model for choices
  - model fit: predictive probability
- ML / MAP
  - parameter inference
  - prior inferred from all joint data
- Empirical prior
  - Infer with approximate EM
  - second level analysis:
    - priors
    - individual posterior parameters
- Model comparison
  - Normal-inverse Gamma -> Gaussian mixture

Softmax likelihood

$$\mathcal{L}(\theta) = p(\{a_t\}_{t=1}^T | \{s_t\}_{t=1}^T, \{r_t\}_{t=1}^T, \theta) = \prod_{t=1}^T p(a_t | s_t, r_{1\ldots t-1}, \theta)$$

- log is easier:

$$\begin{aligned} \log \mathcal{L}(\theta) &= \sum_{t=1}^T \log p(a_t | s_t, r_{1\ldots t-1}, \theta) \\ &= \sum_{t=1}^T \left[\beta \mathcal{Q}_t(a_t, s_t) - \log \sum_{a'} e^{\beta \mathcal{Q}_t(a', s_t)} \right] \end{aligned}$$

ML by gradient ascent

$$\frac{\log \mathcal{L}(\theta)}{d\beta} = \sum_{t=1}^{T} \left[\mathcal{Q}_t(a_t, s_t) - \frac{\sum_{a'} e^{\beta \mathcal{Q}_t(a', s_t)}}{\sum_{a''} e^{\beta \mathcal{Q}_t(a'', s_t)}} \mathcal{Q}(a', s_t) \right]$$

ML by gradient ascent

$$\frac{\log \mathcal{L}(\theta)}{d\beta} = \sum_{t=1}^{T} \left[\mathcal{Q}_t(a_t, s_t) - \frac{\sum_{a'} e^{\beta \mathcal{Q}_t(a', s_t)}}{\sum_{a''} e^{\beta \mathcal{Q}_t(a'', s_t)}} \mathcal{Q}(a', s_t) \right]$$

$$= \sum_{t=1}^{T} \left[\mathcal{Q}_t(a_t, s_t) - \sum_{a'} p_t(a'|s_t) \mathcal{Q}_t(a', s_t) \right]$$

ML by gradient ascent

$$\frac{\log \mathcal{L}(\theta)}{d\beta} = \sum_{t=1}^{T}\left[\mathcal{Q}_t(a_t,s_t) - \frac{\sum_{a'} e^{\beta\mathcal{Q}_t(a',s_t)}}{\sum_{a''} e^{\beta\mathcal{Q}_t(a'',s_t)}}\mathcal{Q}(a',s_t)\right]$$

$$= \sum_{t=1}^{T}\left[\mathcal{Q}_t(a_t,s_t) - \sum_{a'} p_t(a'|s_t)\mathcal{Q}_t(a',s_t)\right]$$

$$\frac{\log \mathcal{L}(\theta)}{d\alpha} = \beta\sum_{t=1}^{T}\left[\frac{d\mathcal{Q}_t(a_t,s_t)}{d\alpha} - \sum_{a'} p_t(a'|s_t)\frac{d\mathcal{Q}(a',s_t)}{d\alpha}\right]$$

ML by gradient ascent

$$\frac{\log \mathcal{L}(\theta)}{d\beta} = \sum_{t=1}^{T}\left[\mathcal{Q}_t(a_t, s_t) - \frac{\sum_{a'} e^{\beta \mathcal{Q}_t(a', s_t)}}{\sum_{a''} e^{\beta \mathcal{Q}_t(a'', s_t)}} \mathcal{Q}(a', s_t)\right]$$

$$= \sum_{t=1}^{T}\left[\mathcal{Q}_t(a_t, s_t) - \sum_{a'} p_t(a'|s_t)\mathcal{Q}_t(a', s_t)\right]$$

$$\frac{\log \mathcal{L}(\theta)}{d\alpha} = \beta \sum_{t=1}^{T}\left[\frac{d\mathcal{Q}_t(a_t, s_t)}{d\alpha} - \sum_{a'} p_t(a'|s_t)\frac{d\mathcal{Q}(a', s_t)}{d\alpha}\right]$$

$$\frac{d\mathcal{Q}_t(a_t, s_t)}{d\alpha} = (1-\alpha)\frac{d\mathcal{Q}_{t-1}(a_t, s_t)}{d\alpha} - \mathcal{Q}_{t-1}(a', s_t) + r_t$$

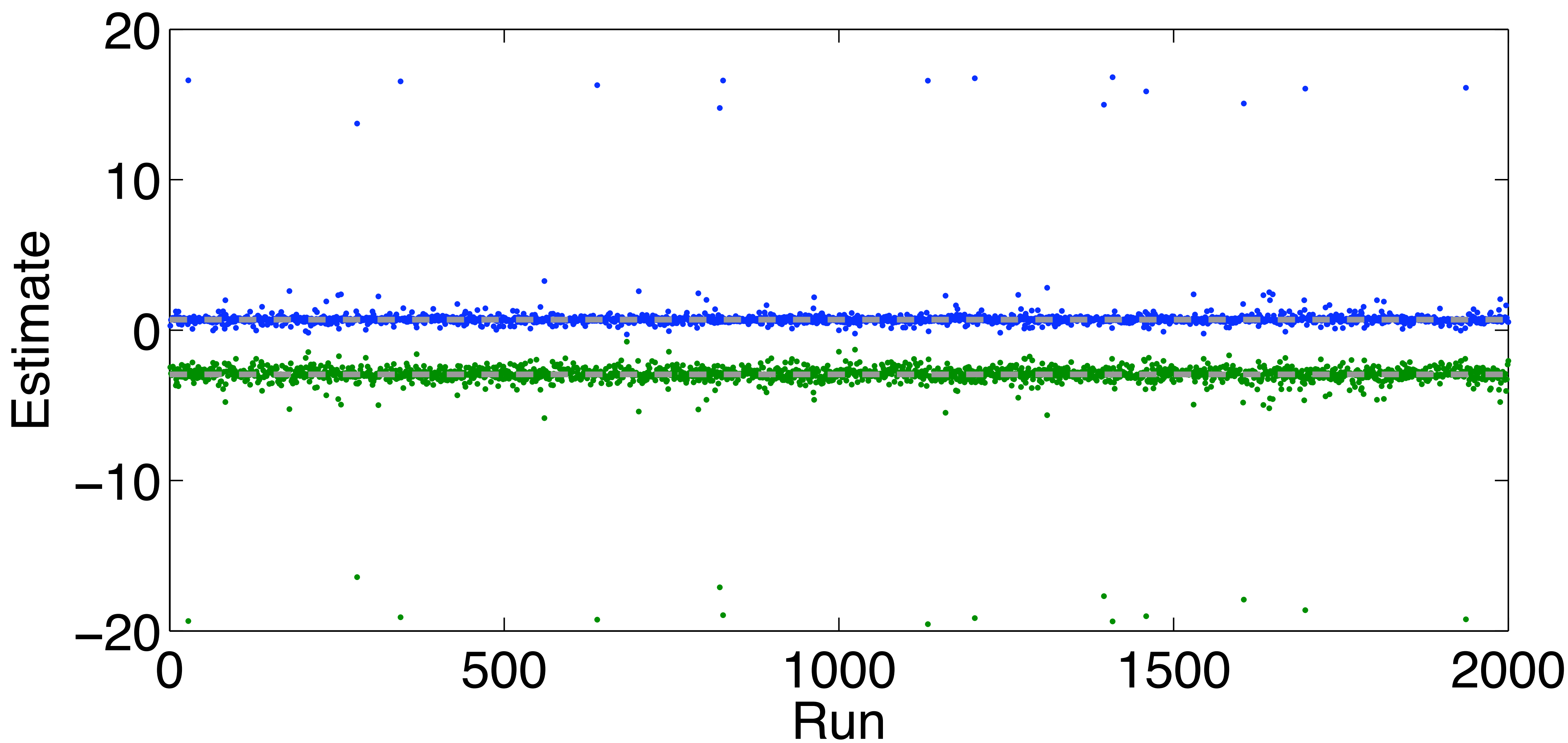
Transforming variables

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

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

ML can be noisy

$$\mathcal{L}(\beta = 10) \approx \mathcal{L}(\beta = 100)$$



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

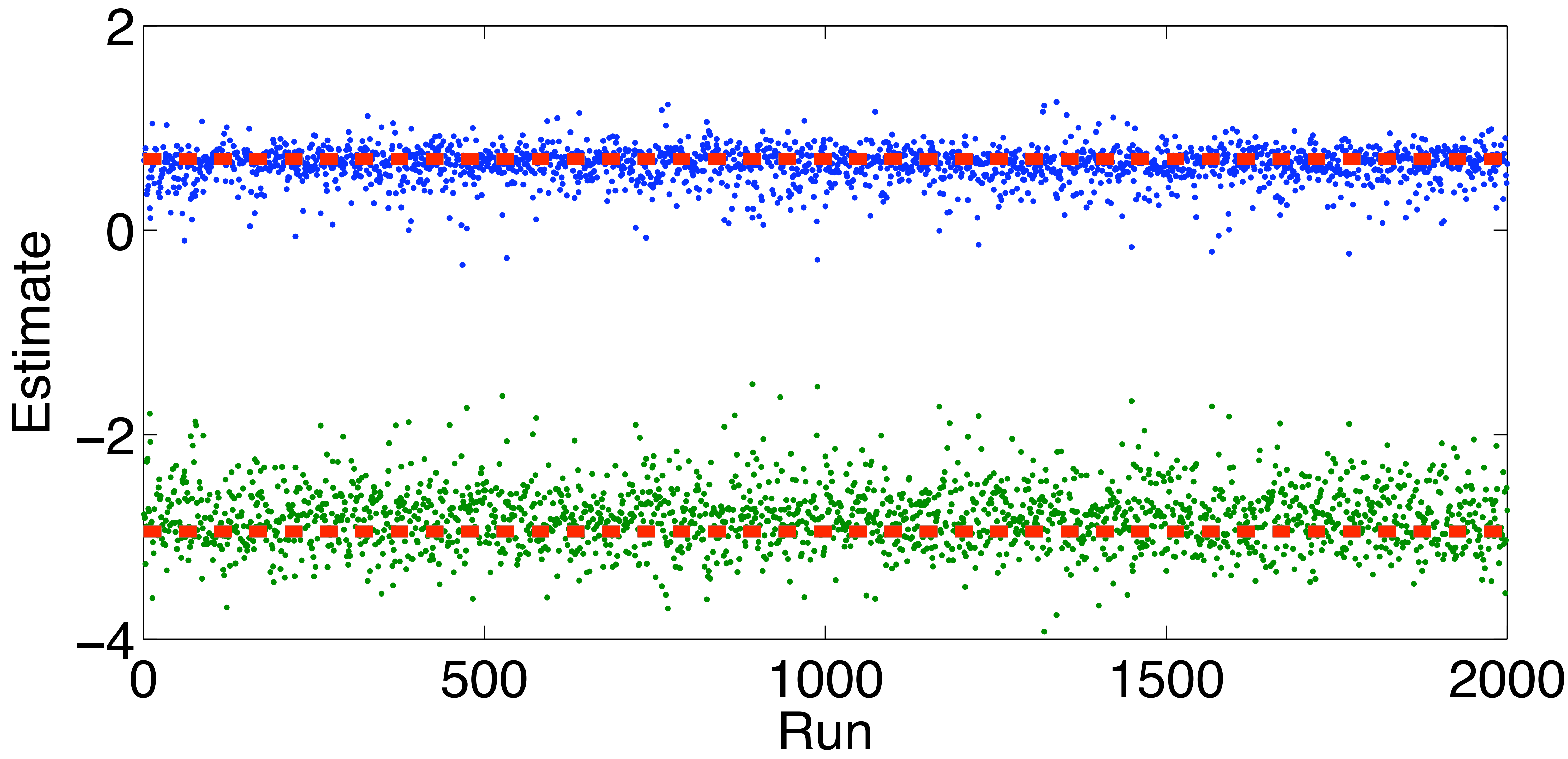
Maximum a posteriori estimate

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

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

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

Maximum a posteriori estimate



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

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

Overview

- Formulate probabilistic model for choices
  - model fit: predictive probability
- ML / MAP
  - parameter inference
  - prior inferred from all joint data
- Empirical prior
  - Infer with approximate EM
  - second level analysis:
    - priors
    - individual posterior parameters
- Model comparison
  - Normal-inverse Gamma -> Gaussian mixture

Estimating the hyperparameters

- What should the hyperparameters be?

$$\log \mathcal{P}(\theta) = \mathcal{L}(\theta) + \log \underbrace{p(\theta)}_{=p(\theta|\zeta)} + const.$$

- Empirical Bayes: set them to ML estimate

$$\hat{\zeta} = \operatorname*{argmax}_{\zeta} p(\mathcal{A}|\zeta)$$

- where we use all the actions by all the *k* subjects

$$\mathcal{A} = \{a^k_{1\ldots T}\}_{k=1}^{K}$$

Estimating the hyperparameters

- Need to integrate out individual parameters:

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

- Standard problem, apply EM

EM with Laplace approximation

$$
\begin{aligned}
\text{E step:} \quad q_k(\theta) &= \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k) \\
\mathbf{m}_k &= \underset{\theta}{\operatorname{argmax}}\, p(\mathbf{a}^k|\theta)p(\theta|\zeta_i) \\
\mathbf{S}_k^{-1} &= \left.\frac{\partial^2 p(\mathbf{a}^k|\theta)p(\theta|\zeta_i)}{\partial\theta^2}\right|_{\theta=\mathbf{m}_k} \\
\text{M step:} \quad \zeta_{i+1}^{\mu} &= \frac{1}{K}\sum_k \mathbf{m}_k \\
\zeta_{i+1}^{\nu^2} &= var(\mathbf{m}_k)
\end{aligned}
$$

Priors and 2nd level analysis

- Priors over parameters
  - can do this for subgroups

$$p(\theta|\hat{\zeta})$$

- Posterior parameter estimates
  - do classical second level analyses
  - can use Hessians as weights

$$\begin{aligned} \text{point estimates} \quad \hat{\theta}^k = & \quad \mathbf{m}^k \\ \text{precisions} \quad & \quad \mathbf{S}^k \end{aligned}$$

Overview

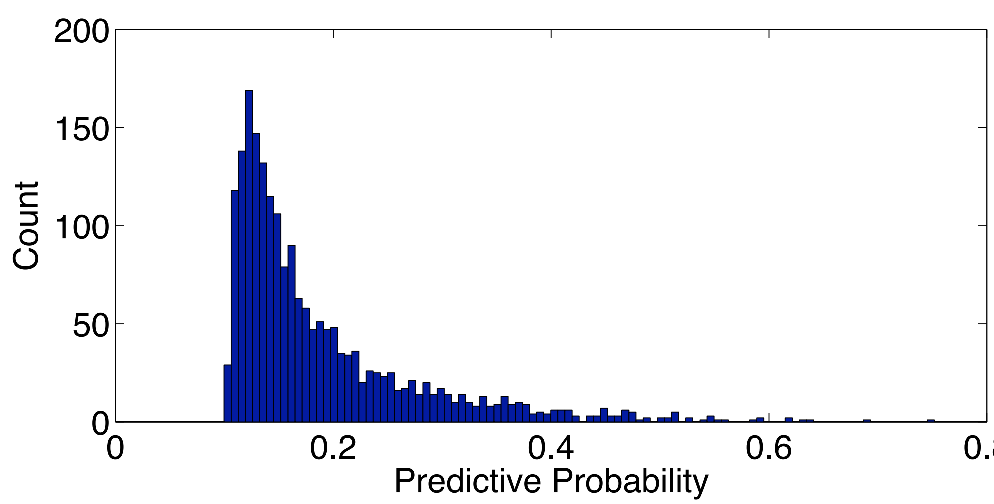
- Formulate probabilistic model for choices
  - model fit: predictive probability
- ML / MAP
  - parameter inference
  - prior inferred from all joint data
- Empirical prior
  - Infer with approximate EM
  - second level analysis:
    - priors
    - individual posterior parameters
- Model comparison
  - Normal-inverse Gamma -> Gaussian mixture

Model fit: predictive probabilities

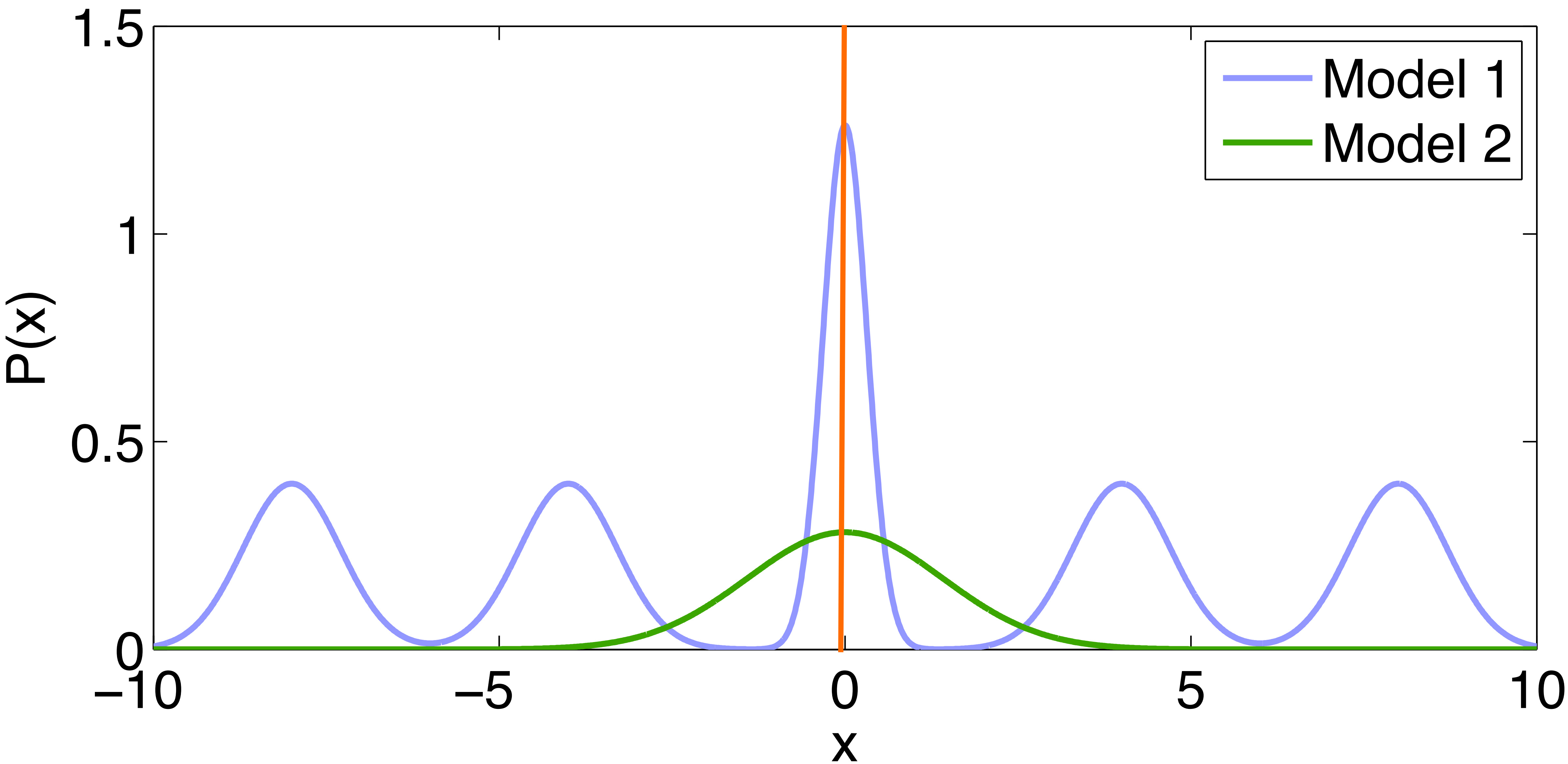
- How well does the model do?
  - choice probabilities:

$$
\begin{aligned}
\mathbb{E}p(correct) &= e^{\mathcal{L}(\hat{\theta})/K/T} \\
&= e^{\log p(\mathcal{A}|\theta)/K/T} \\
&= \left(\prod_{k,t=1}^{K,T} p(a_{k,t}|\theta) \right)^{\frac{1}{KT}}
\end{aligned}
$$

  - typically around 0.65-0.75 for 2-way choice
  - for 10-armed bandit example:



Model comparison



Model comparison

- Penalise for overly broad predictions

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

- where we can simplify a bit

$$\begin{aligned} p(\mathcal{A}|\mathcal{M}_1) &= \int d\zeta \int d\theta \, p(\mathcal{A}|\theta) \, p(\theta|\zeta) \, p(\zeta|\mathcal{M}) \\ &= \int d\theta \, p(\mathcal{A}|\theta) \, p(\theta|\mathcal{M}) \end{aligned}$$

Model comparison

- Prior form

$$p(\theta|\mathcal{M}) = \int d\zeta\, p(\theta|\zeta) \underbrace{p(\zeta|\mathcal{M})}_{p(\mu,\nu^2|\mathcal{M})}$$

- straightforward option is conjugate prior, in this case Normal-inverse Gamma

$$p(\mu,\nu^2|\mathcal{M}) = \frac{b^a}{\Gamma(a)}\left(\frac{1}{\nu^2}\right)^{a+1}\exp\left(-\frac{b}{\nu^2}\right)\frac{s}{\sqrt{2\pi\nu}}\exp\left(-\frac{(\mu-m)^2}{2\nu^2/s^2}\right)$$

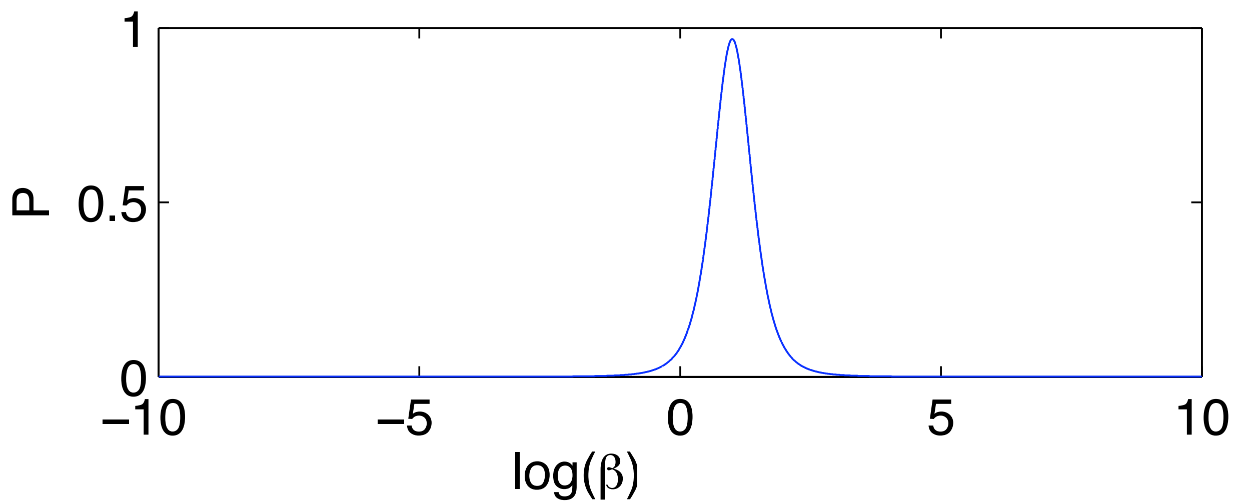
- which gives us a Gaussian scale mixture

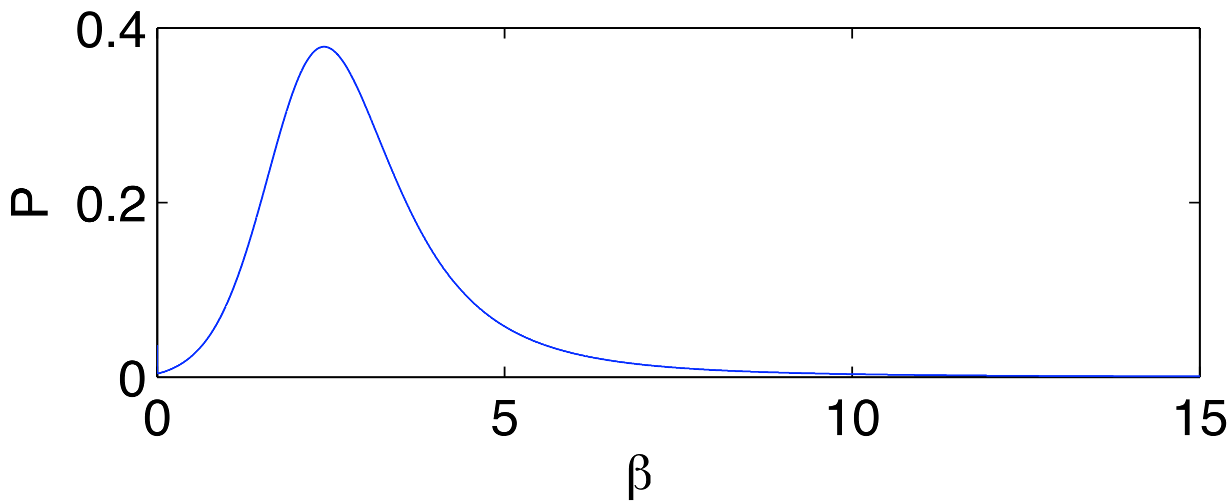
$$p(\beta|\mathcal{M}) = \frac{\Gamma(a+\frac{1}{2})}{\Gamma(a)}\frac{b^a}{\sqrt{2\pi(1+1/s^2)}}\left(\frac{(\beta-m)^2}{2(1+1/s^2)}+b\right)^{-(a+\frac{1}{2})}$$

For a simple RW model

Prior on transformed variable

Prior on original variable





- Evaluate integral by sampling

$$
\begin{aligned}
p(\mathcal{A}|\mathcal{M}_1) &= \int d\theta\, p(\mathcal{A}|\theta)\, p(\theta|\mathcal{M}) \\
&\approx \frac{1}{N}\sum_i p(\mathcal{A}|\theta_i); \qquad \theta_i \sim p(\theta|\mathcal{M})
\end{aligned}
$$

Overview

- Formulate probabilistic model for choices
  - model fit: predictive probability
- ML / MAP
  - parameter inference
  - prior inferred from all joint data
- Empirical prior
  - Infer with approximate EM
  - second level analysis:
    - priors
    - individual posterior parameters
- Model comparison
  - Normal-inverse Gamma -> Gaussian mixture