Reinforcement Learning and Animal Learning

Chris Watkins

Royal Holloway, University of London

Is RL an adequate account of animal learning?

1. Knowledge representation: does RL produce the right knowledge?
2. Large state spaces: ‘reward state’
3. Exploration: RL is under-specified
4. Generalisation by NN interpolation is correct discovery procedure?
5. Initial conditions greatly affect RL
6. Final behaviour is (often) innate: is innate specification of a reward function plausible?
7. Specific examples of non-reward-based learning: instinctual drift: are these exceptions, or fundamental?
8. Hyperbolic discounting: preferences change over time

The apparent promise of RL:

‘Entry-level learning’: actor-critic or Q learning

- acquire optimal behaviour without long-term memory or prediction of state transitions
- policy gradient: optimise policy without V or Q

“Obvious” research program:

- use function approximation for V or Q over large state spaces
- learn state transition models and combine with RL for upgraded learning

(Took 25 years! Now a vast number of different ideas being tried…)

Apparently attractive as a theory of animal learning:

- entry-level Q learners could progressively evolve upgrades, acquiring larger state-space, look-ahead with learned models, better rewards…

Knowledge representations of RL

- Policy
- Value function / Q / Advantages
- Forward simulation
- State space

Policy

- tells agent what it actions it may perform in a given state

- does not say *why* some actions are recommended and others are not

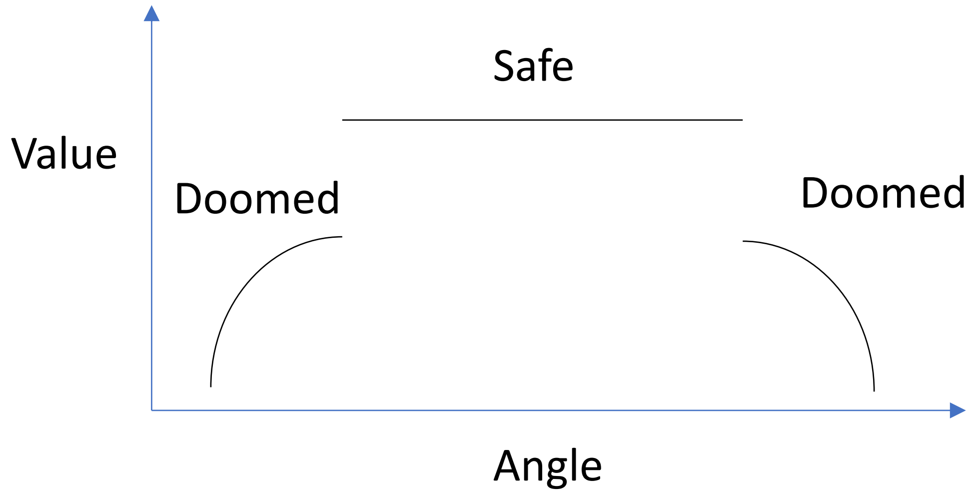
- an agent following a policy has no idea whether its policy is working or not.

Value function

- Typically discontinuous
- Discontinuous gradient
- Semantically diverse
  - Different components of value are combined
  - Joining of different plans
- Value relates to only one plan for each state !

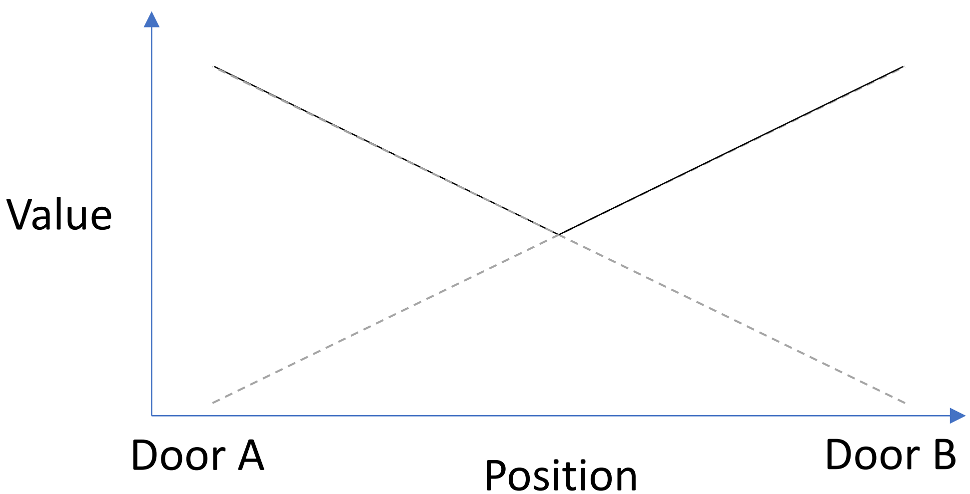
- Interpolation of recursively estimated value is not a powerful discovery procedure.
- Value function is final result of behavioural optimization; much information has been discarded
- Learning a correct value function over a *large* state space is … unlikely. Is an incorrect value function the best intermediate form of knowledge?

Cross section through value function of pole balancer



Value functions are typically discontinuous !

Value function for leaving the lecture theatre



Value functions typically have discontinuous gradient

For each position the value function relates to only one plan.

The other plan is forgotten.

Inappropriate optimisation

RL tries to compute optimal policy and value **directly** from a forward simulator (or experience)

- No useful intermediate knowledge produced
- No deep understanding of environment is found
- Good performance under controlled conditions may be possible, but no reason to expect V to be correct in unvisited regions

The Curse of Dimensionality: how does a state space become large?

- Representing everything: complex dynamics
- Goals as part of state
- Implicit prediction
  - Value / action may depend on future events; the *prediction* of these events may be lower-dimensional than the information necessary to make a prediction
- Path dependency of reward
  - Tying a knot
  - Collecting a shopping-list of items in a supermarket
  - ‘reward state’ can be larger than ‘dynamic state’

Return: an artificial construct

Some state-action sequences (paths) are more desirable than others

We may define a function over **paths** called “utility” or “return” that indicates the desirability of a path

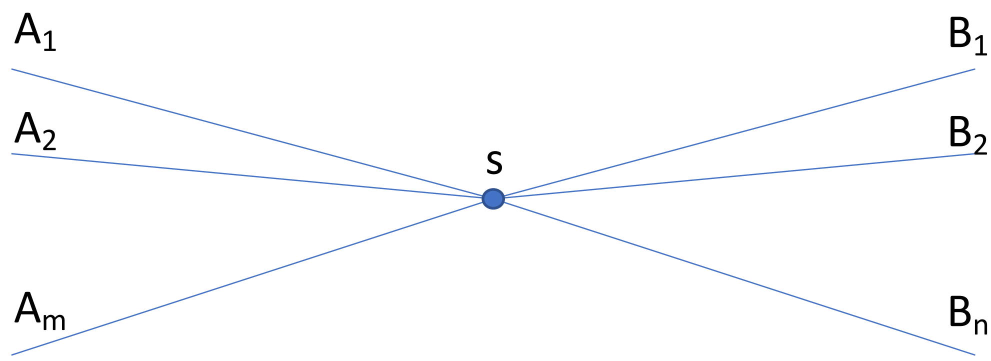
It is convenient if this function is additive so that we may decompose utility of a path into the sum of utilities of its segments: then we can do dynamic programming.

If a utility function on paths is not additive, need to expand the state space to make it additive.

This new component of state might be called ‘**return state**’

Achieving additivity of return

Paths through state space that all pass through state s



Consider mn paths through s. Specify returns of these paths arbitrarily (ie return is path-dependent).

But additivity of return requires $r(A_iB_j) = r(A_i) + r(B_j)$ for all i, j.

There are mn whole-path returns, but only m+n half-path returns.

Therefore in general, to achieve additivity we must increase the size of the state space, so that states include path information.

Exploration

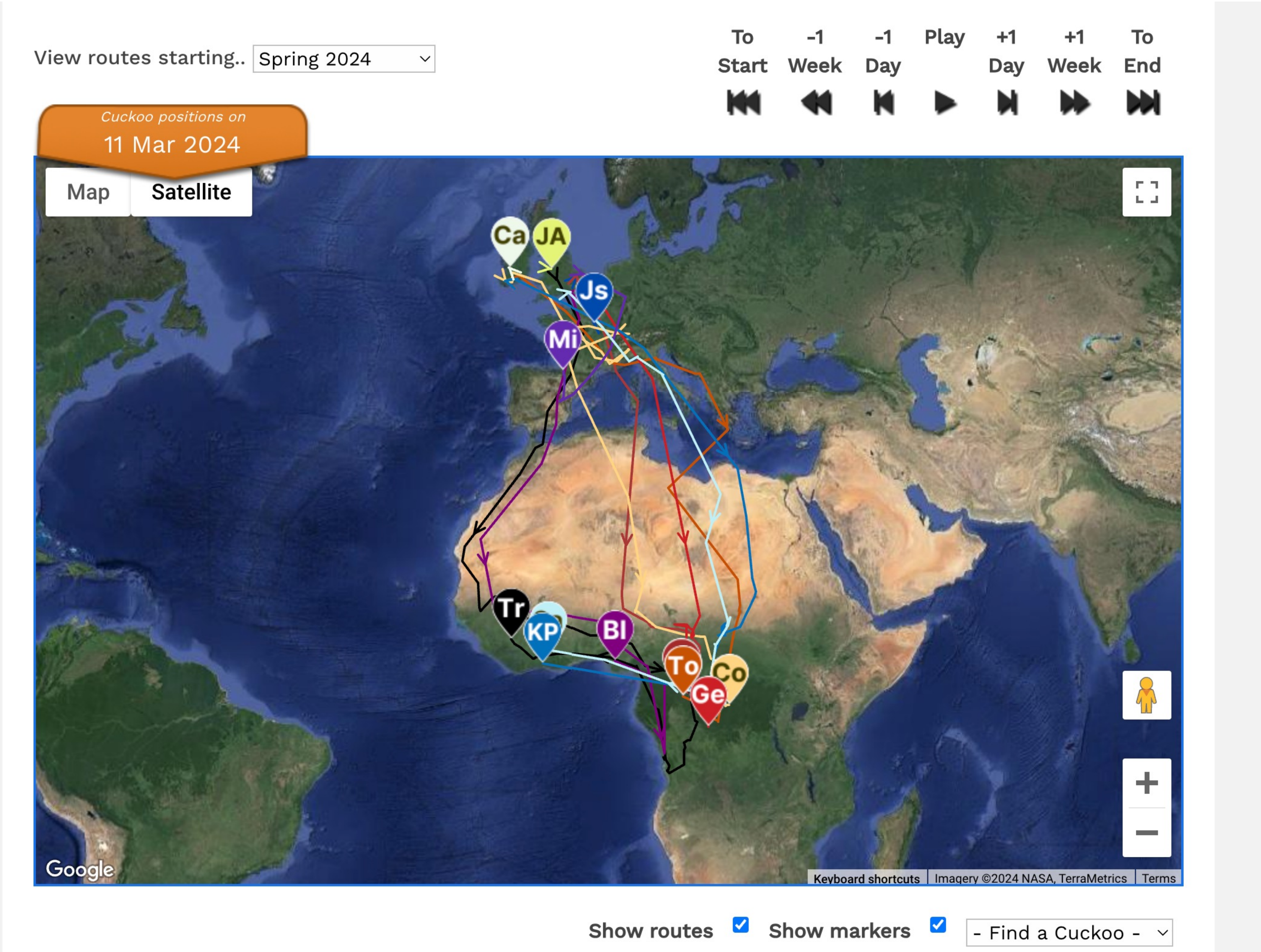
Exploration critically important, especially in large state spaces

Only a tiny part of state-space visited: agent must visit the right part!

Specifying exploration is an active research direction.

Outstanding success:

exploration by self-play in adversarial games



https://www.bto.org/cuckoos

Speculative

Optimality or Autonomy?

Should the design aim be:

- **Optimality**: efficient behaviour in a particular problem under controlled conditions?

- **‘Autonomy’**: adequate behaviour in as wide a range of states as possible?

Does autonomy need causal understanding?

Sheep are not reliably autonomous

They get into dangerous states that they cannot get out of.

Also, no general concept of 'reverse'

Speculative

What would a better theory look like?

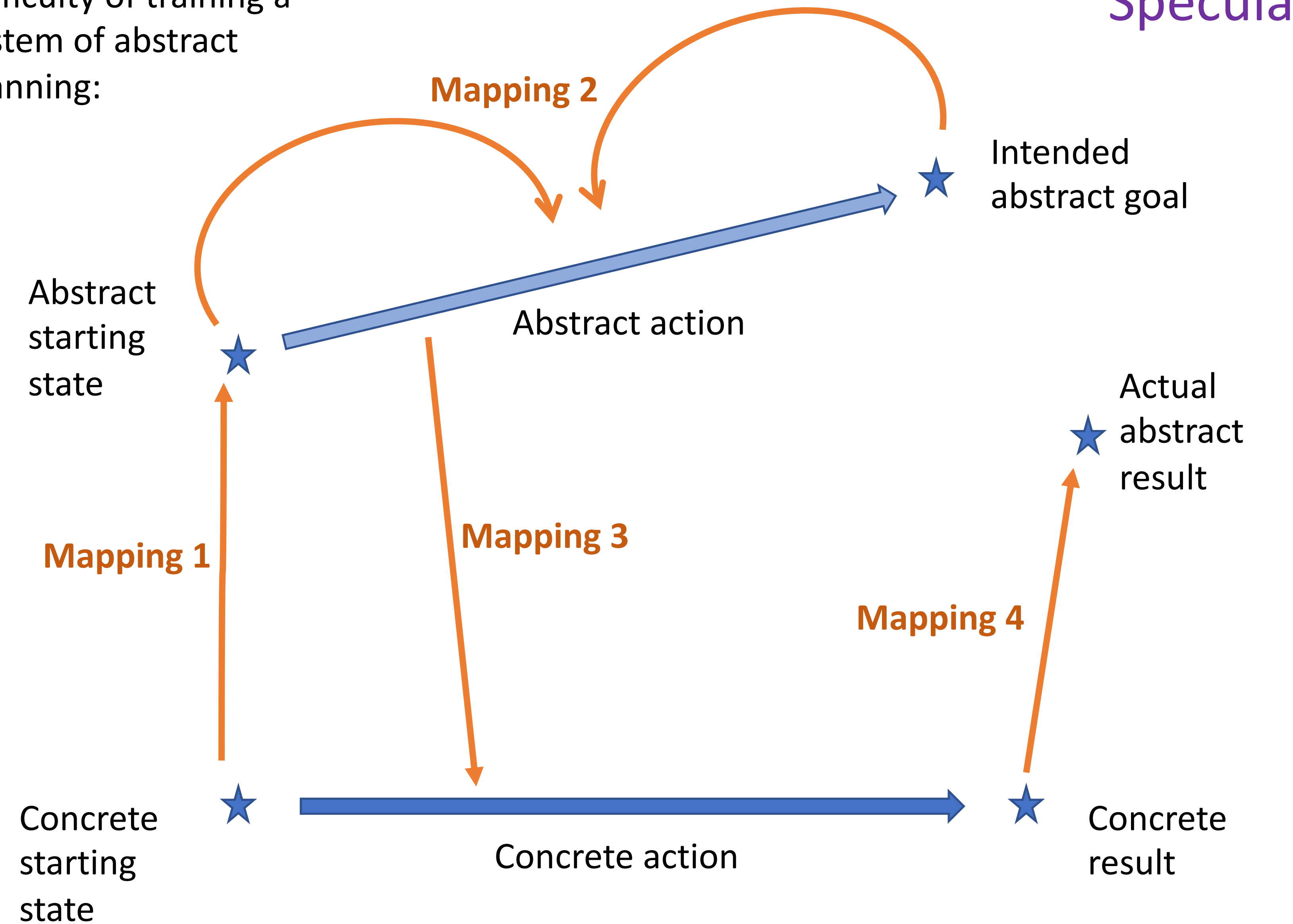
Predicting only the value of a state is restrictive.

Need to make richer abstract predictions of outcomes of feasible plans:

- ‘abstract state’ should be defined so that feasible ‘abstract actions’ can lead to other ‘abstract states’

Difficulty of training a system of abstract planning:

Speculative



Actual result ≠ intended goal. Which mapping is wrong? How to update?

Conclusions

Current RL paradigm is limited…

- Inappropriate optimisation gives brittle solutions
- Interpolation of V or Q is a weak discovery procedure
  - Law-like understanding of effects of actions is desirable
- Optimality or autonomy?
- Abstraction for planning seems hard to train…