11.01.2023

A comparison of reinforcement learning models of human spatial navigation

Article published by Quilang He, Jany Ling Liu, Lou Eschapasse, Elisabeht H. Beveridge & Thackery I. Brown

in *Scientific Reports* (12, 2022)

Presentation by Simon Heuschkel & Peter Wolters

The plan for today

Introduction

- Little Recap: What is Reinforcement Learning? Model-based vs. Model-free RL
- Goal of the study
- The experimental paradigm: What did the participants had to do?

Different Reinforcement Learning models:

- TD(0)-Agent
- TD(λ)-Agent
- TD(1)-Agent
- Model-based Agent
- Mixed Agent

-break-

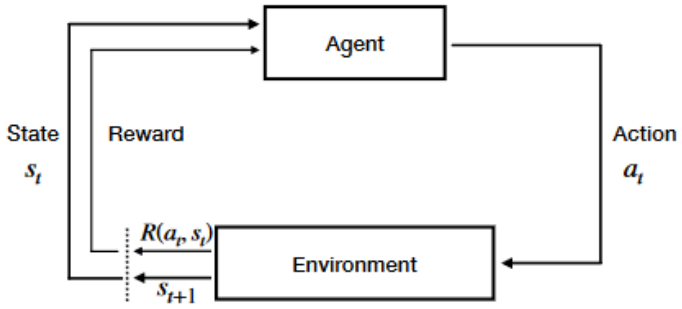
Results:

- How to fit the models? BIC
- What models describe human behavior the best?

Discussion

Short Recap - Reinforcement Learning

RL setup



Sutton and Barto (2018 [1998])

RL as framework

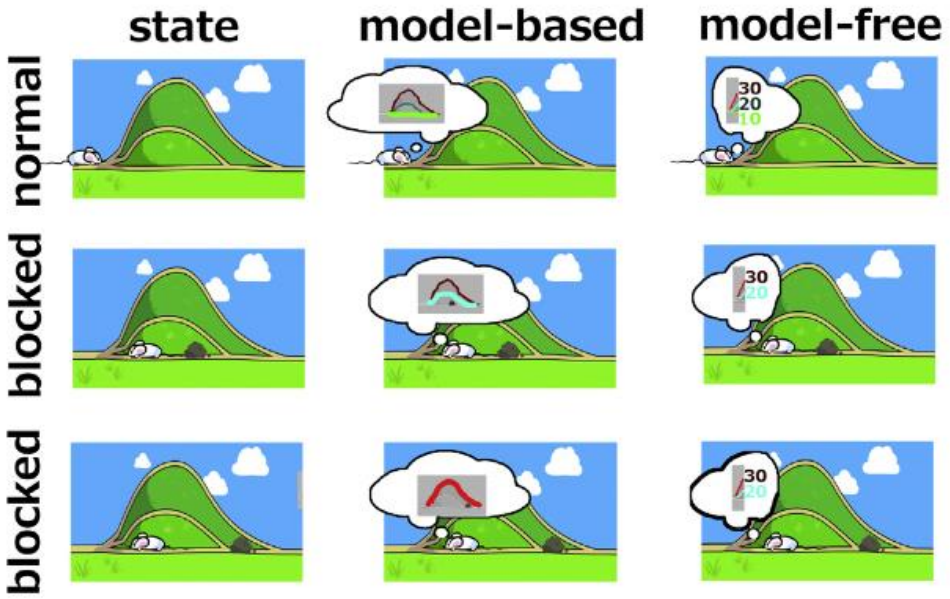
Normative:

- What *should* an agent work to improve its behavior

Descriptive:

- How *does* an agent work, and which mechanisms describe behavior

Model based vs. free RL



Dolan, Dayan (2013)

Objective of study

- All about navigation strategy
- 3D “survival like” navigation task
- RL models and parameters to gain insights into human navigation
  - RL as: function(prior experience) = navigation strategy
  - Insights into cognitive mechanisms
- 5 different RL Models
  - 3 – Model-free
  - 1 – Model-based
  - 1 – Hybrid-Model

Experiment – Dual Solution Task

Task: Find one objective in the environment per trial

- 3 possible objectives (apple, banana, watermelon)
- Only current objective is visible

1. Fixed phase

- 3 trials per objective
  - Same sequence across participants
- Fixed starting position & orientation

→ Like going to grocery store

→ Don’t get too familiar

2. Random phase

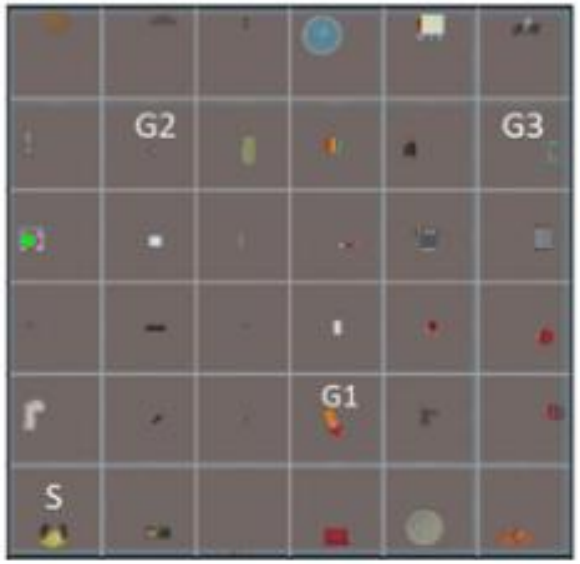
- 72 trials
- Random starting location & orientation
- Random Objectives sequence

→ Transfer knowledge from previous trials

→ Continue learning in probabilistic environment

Experiment – Environment

A



B

C

- 6x6 Grid
- Every room has a reference object (here: robot & car)
- No global landmark

Experiment – Task Measurements

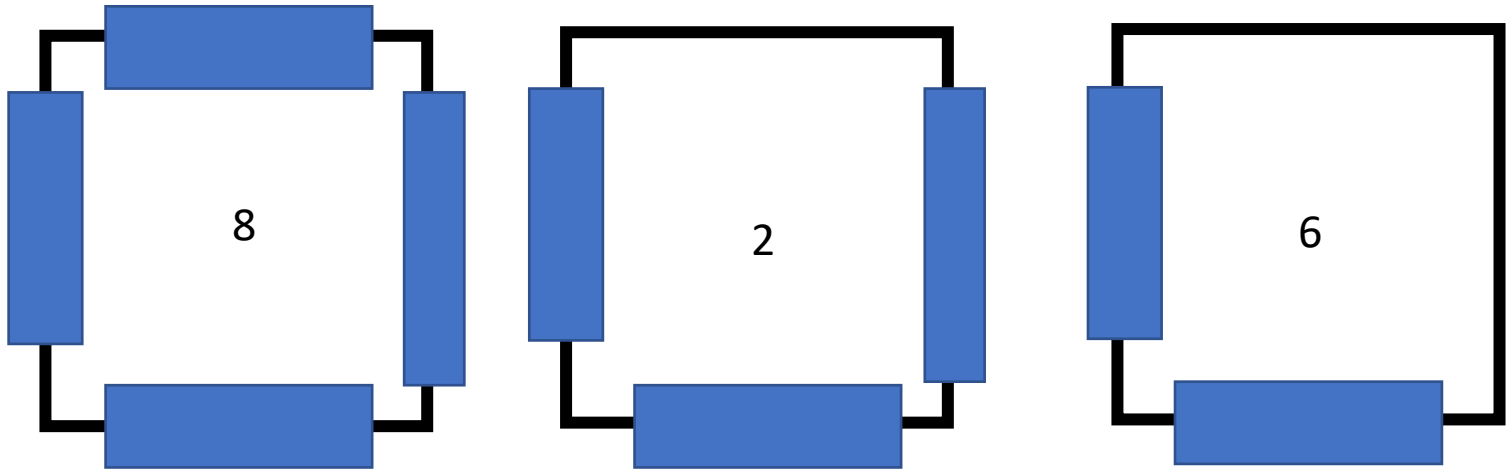
- Way finding efficiency
  - Excessive distance = (actual traversed distance − optimal distance)/ optimal distance
  - ED = 0 → optimal distance
  - ED = 1 → 2 * the optimal distance
- 114 participants for data analysis
- Now we're going to look at the models

TD(0)-Agent

- Learns action values/Q-values Q(s,a)

States (s) = different rooms (6 x 6 = 36 states)
Actions (a) = which door to choose (4, 3, or 2 actions per room

S=1	2	3	4	5	6
7	8	…			
			Goal		
Start					



-> in total 4*4*4 + 4*4*3+ 4*2 = 120 different Q-Values

-> The Q-Value represents how favorable the action a in state s is. (Higher Q-value -> more favorable action; Q-value = future reward)

→ How are action values learned?/How are action chosen?

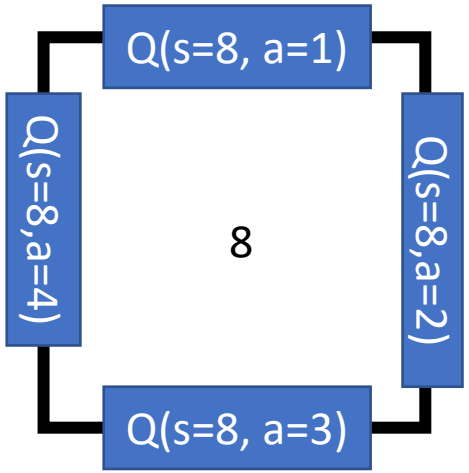
How are action chosen based on action values Q(s,a)?

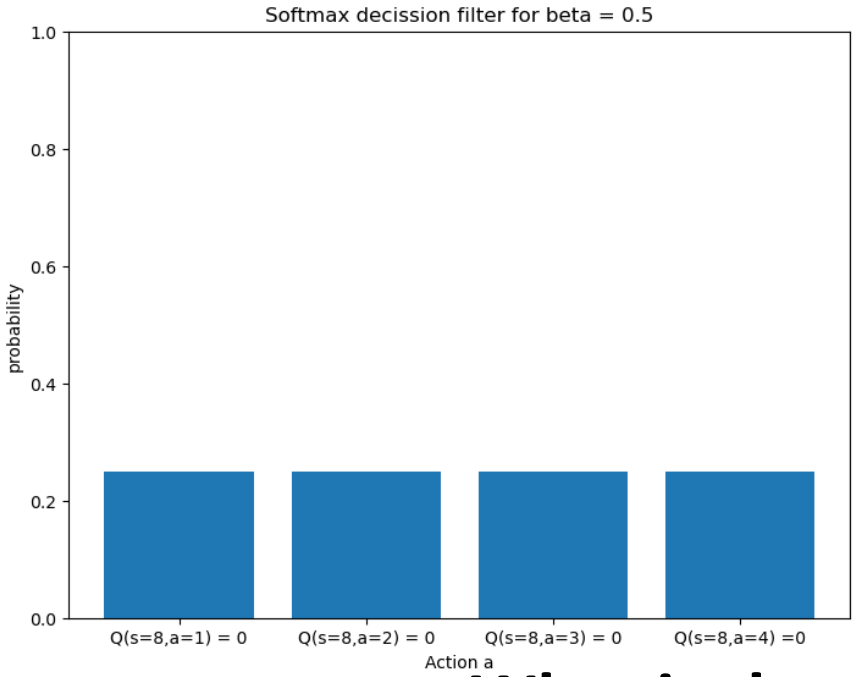
TD(0)-Agent

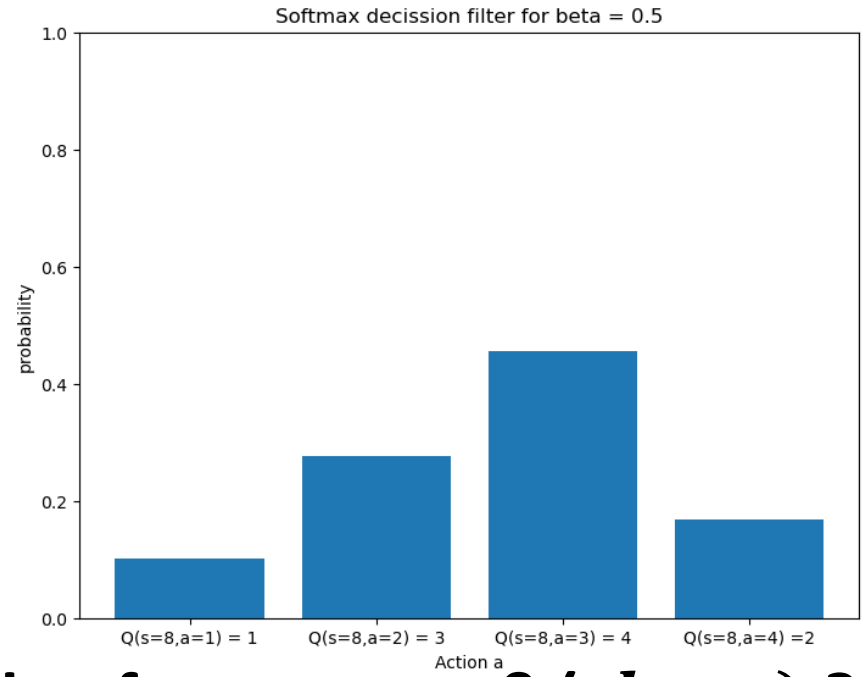
To obtain an action from an action value, we

can use the Softmax function to obtain a policy:

$$\text{Definition: } P(s, action = a) = \frac{\exp(\theta \times Q(s, action=a))}{\sum_{a_i}^{all\ actions\ of\ state\ s} \exp(\theta x Q(s, action=a_i)}$$

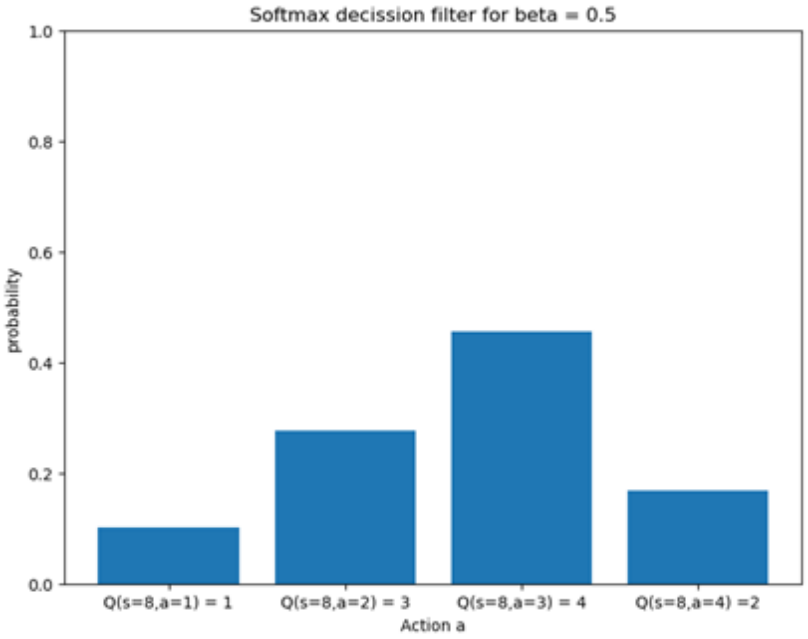


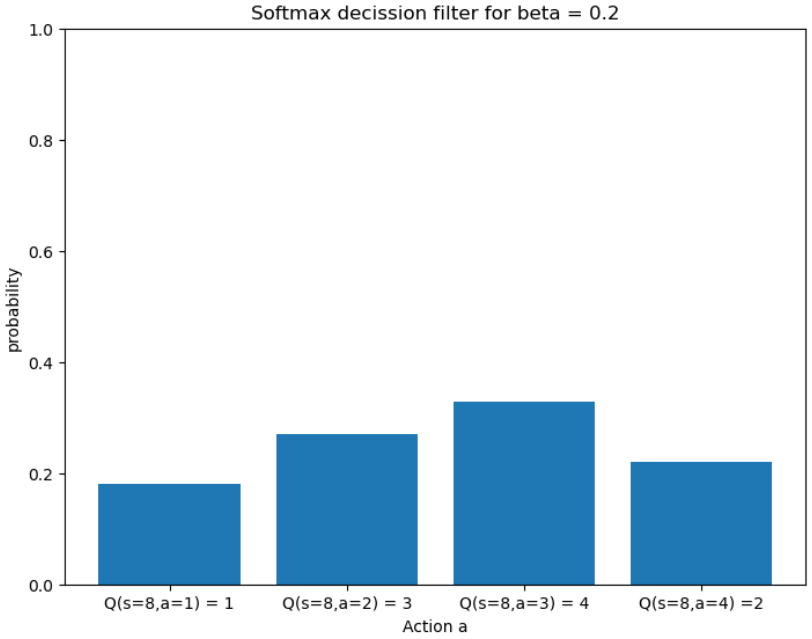


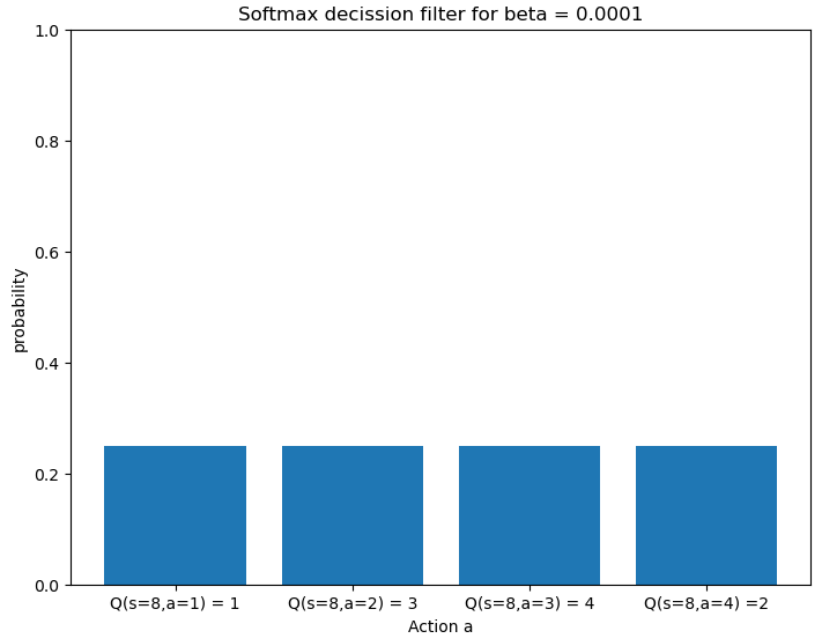


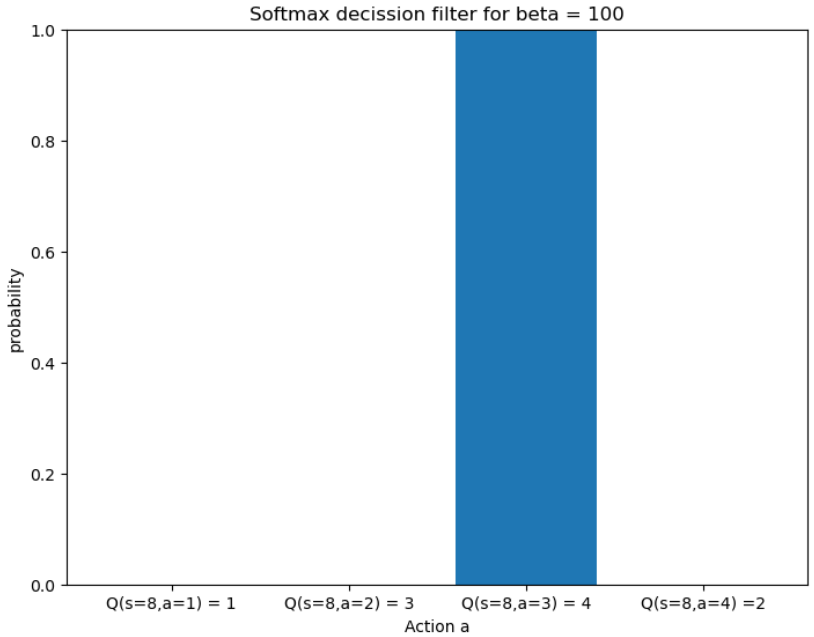
-> What is the role of parameter θ ($theta$) ?

-> What is the role of θ?









TD(0)-Agent

-> θ is called inverse Temperature and can model the

Exploration-Exploitation trait off

TD(0)-Agent

How are the Q-values learned?
TD=Temporal difference learning

S=1	2	3	4	5	
7	8	…			
			Goal		
Start					

Goal: Q(s,a) = total future reward r if the agent takes the action a in state s

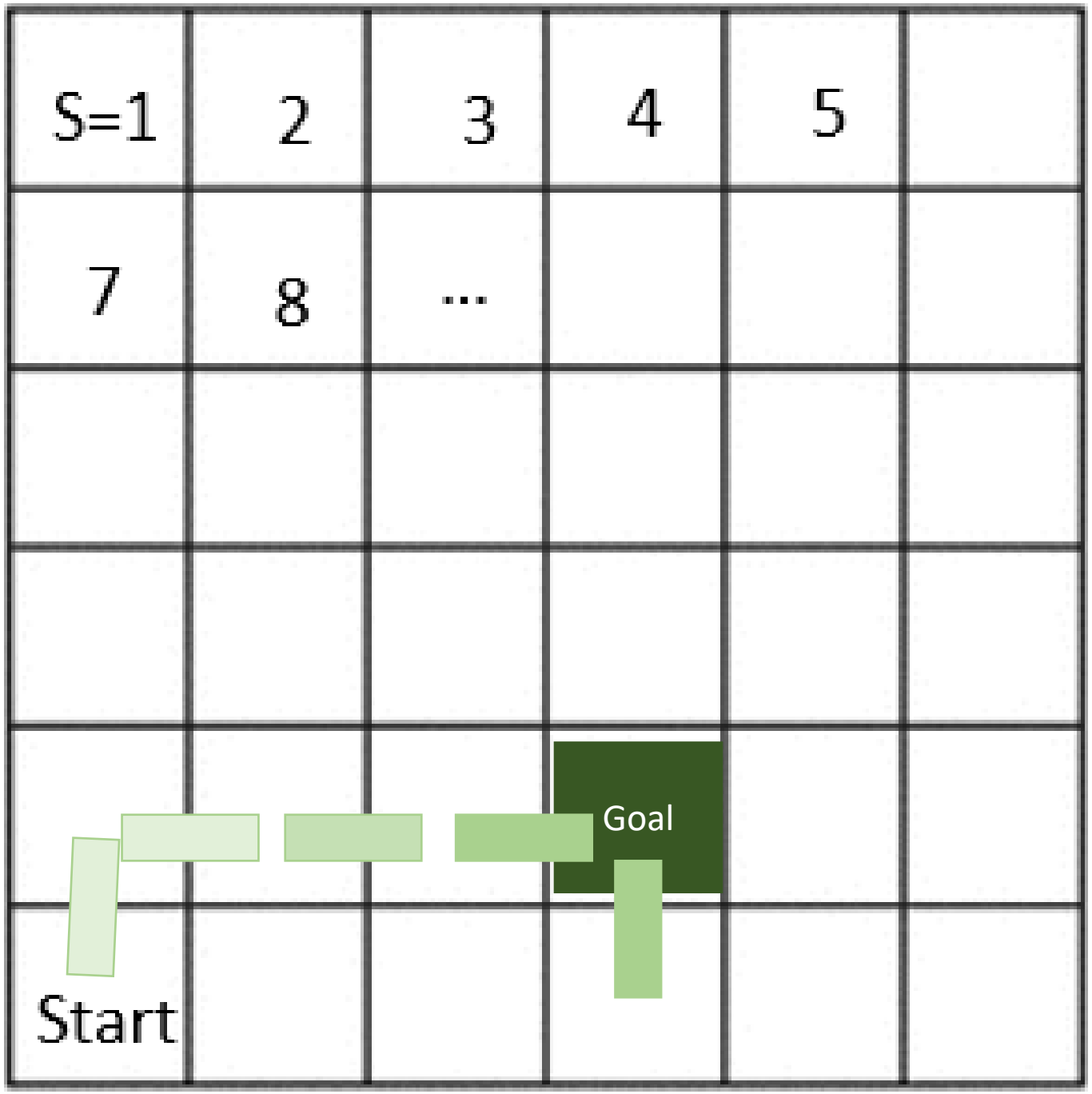
After (t+1) an Action the Q-Value of that Action gets updated with the prediction error delta δ

$$Q(s,a) = Q(s,a) + \alpha\delta$$

$$\delta = \underbrace{r_t + Q(s_{t+1}, a_{t+1})}_{\text{“Actual” future reward}} - \underbrace{Q(s_t, a_t)}_{\text{Predicted future reward}}$$

Time is discretized by the different rooms/states s

α is the learning rate



$$Q(s,a) = Q(s,a) + \alpha\delta$$

$$\delta = r_t + Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$

TD(0)-Agent Summary

- Agent values Q(s,a) are learned
- An agent value Q is updated with the prediction error δ, after the action was performed. (The learning rate α determines how fast an agent learns)
- An action is chosen with the SoftMax function. θ determines the randomeness

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy derived from Q (e.g., ϵ-greedy)
    Take action a, observe r, s'
    $Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]$
    $s \leftarrow s'$;
  until s is terminal

Taken from Sutton et al. *Reinforcement Learning. An Introduction* (MIT Press, 2018)

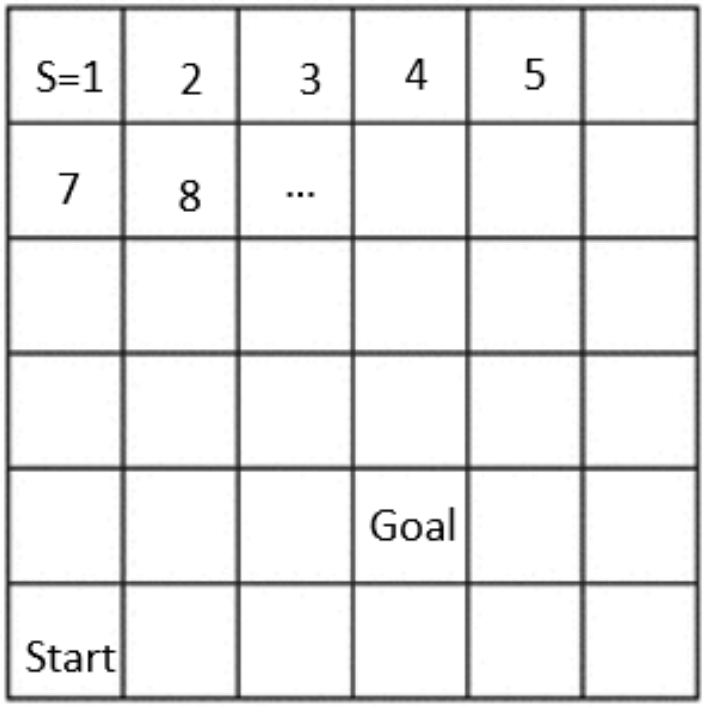
S=1	2	3	4	5	
7	8	...			
			Goal		
Start					

$$Q(s,a) = Q(s,a) + \alpha\delta$$

$$\delta = r_t + Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$

$$P(s, action = a) = \frac{\exp(\theta \times Q(s, action=a))}{\sum_{a_i}^{all\ actions\ of\ state\ s} \exp(\theta x Q(s, action=a_i))}$$

TD(λ)-Agent



Introduce a new concept, the **eligibility trace e(s,a)** with decay parameter λ between 0 and 1

Before each trial e(s,a) = 0

After Action A was performed:

Update e(s,a) **for all** action-state pairs

$$e(s,a) = \lambda\, e_{t-1}(s,a) + 1 \text{(if s=s, and a=A)}$$

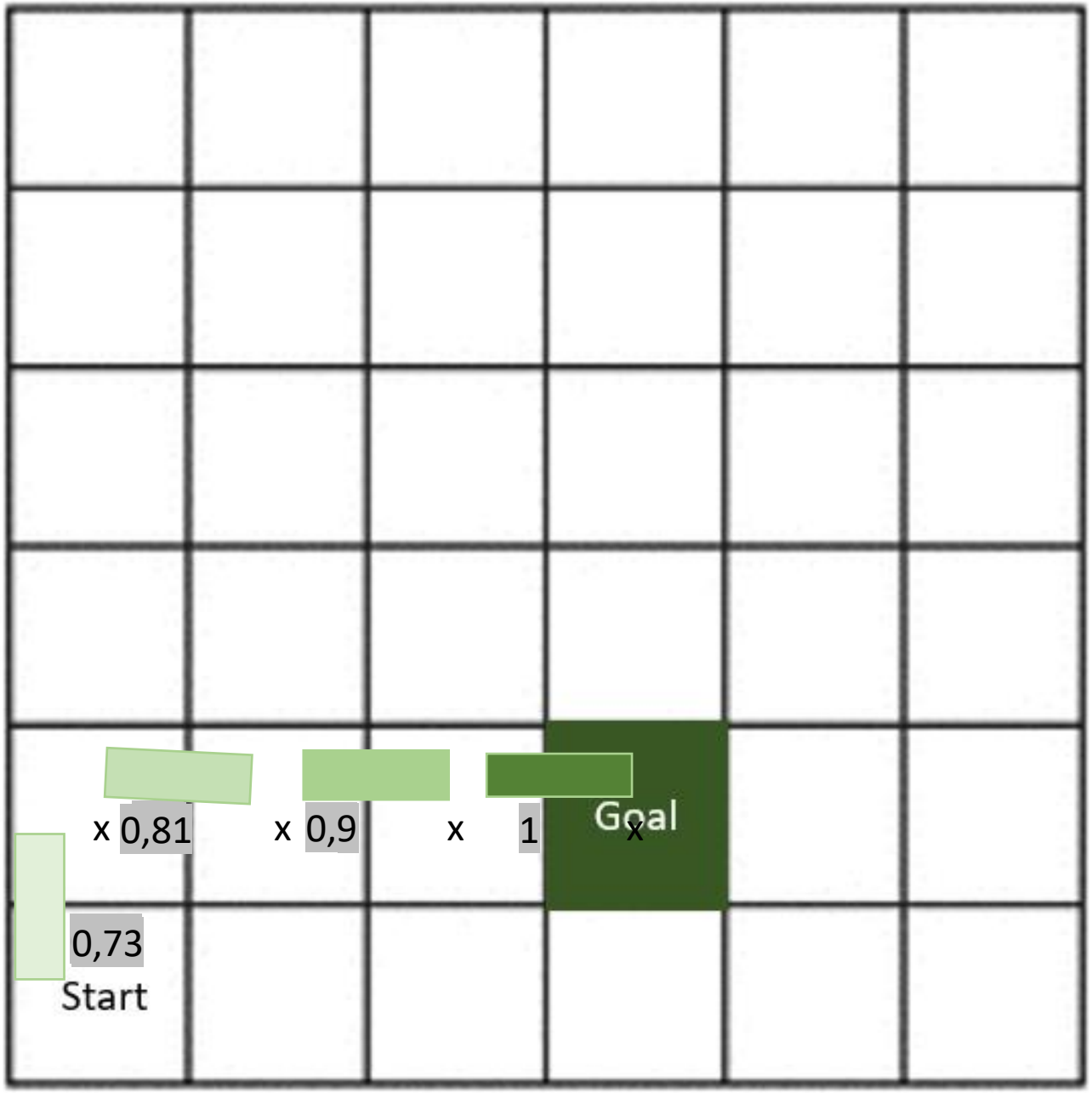
Update Q **for all** action-state pairs

$$Q(s,a) = Q(s,a) + \alpha \delta e(s,a)$$

$$\delta = r_t + Q(s_{t+1}, A_{t+1}) - Q(s_t, A_t)$$

→Frequently visited states are updated stronger

→TD(0) is a special Case with λ=0



Example λ = 0.9

Before each trial e(s,a) = 0

After Action A was performed:

Update Q **<u>for all</u>** action-state pairs

$$Q(s,a) = Q(s,a) + \alpha\delta e(s,a)$$

$$\delta = r_t + Q(s_{t+1}, A_{t+1}) - Q(s_t, A_t)$$

Update e(s,a) **<u>for all</u>** action-state pairs

$$e(s,a) = \lambda\, e_{t-1}(s,a) + 1 \text{(if s=s, and a=A)}$$

TD(1)-Agent

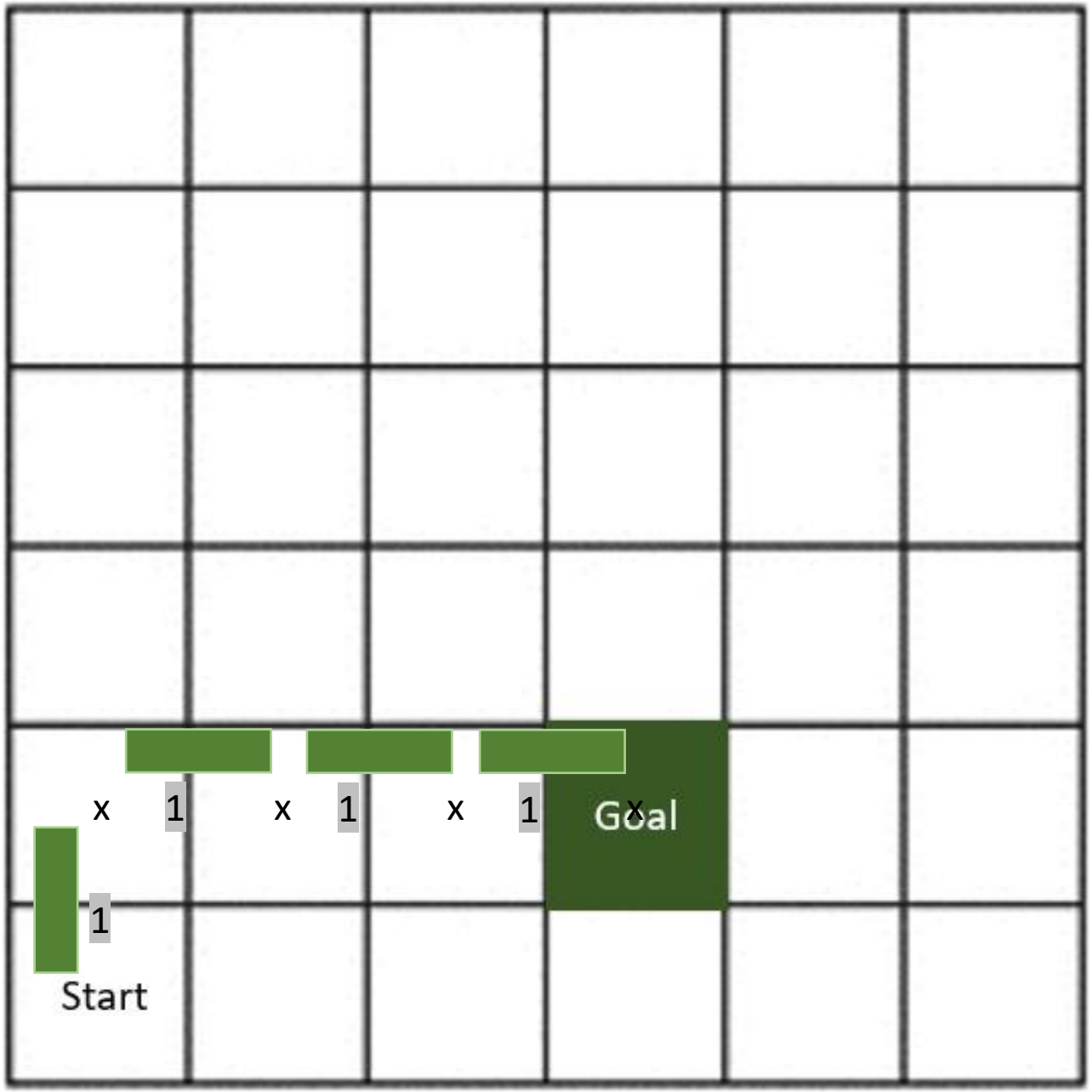
$$e(s,a) = \lambda\, e_{t-1}(s,a) + 1(\text{if s=s, and a=A})$$

- λ = 1 ? Special case too?

„TD(1) was a special case of TD(λ), which forced every visited state to get the same amount of updating regardless of how often they were visited.“

e(s,a) <= 1

$$e(s,a) = \lambda\, e_{t-1}(s,a) + 1(\text{if s=s, and a=A})$$



Example λ = 1

Before each trial e(s,a) = 0

After Action A was performed:

Update Q **<u>for all</u>** action-state pairs

$$Q(s,a) = Q(s,a) + \alpha\delta e(s,a)$$

$$\delta = r_{t+1} + Q(s_{t+1}, A_{t+1}) - Q(s_t, A_t)$$

Update e(s,a) **<u>for all</u>** action-state pairs

$$e(s,a) = \lambda\, e_{t-1}(s,a) + 1 \text{(if s=s, and a=A)}$$

e(s,a) <= 1

TD(0)-Agent TD(1)-Agent T(λ)-Agent

- All model-free RL models use the TD-Error
- Learn action-values Q(s,a) for different state and action pairs (s,a)
- All have a learning rate α and inverse temperature θ
- TD(λ) models memory decay with the parameter λ

- TD(0), and TD(1) -> 2 Parameters: α and θ
- TD(λ) -> 3 Parameters: θ, λ

Model-based Model

- All $Q_{MB} = 0$ in the beginning
- Travers all possible rooms and directions to goal

$$Q_{MB}(s) \leftarrow \sum_a \pi(a|s) + \sum_{s',r} p(s', r|s, a)\left[r + \gamma Q_{MB}(s')\right]$$

Policy:
Explore/ Exploit

Probability for reward and state, given current state & action

$\gamma = 0.8$

- Converge until difference < 0.0001
- All participants had the same perfect map (“Cognitive Map“)
- Parameter → 1 (θ)

Hybrid Model

$$Q_{hybrid} = (1 - \omega)Q_{MF} + \omega Q_{MB}$$

- ω
  - High → More model-based learning
  - Navigational strategy
- Best TD Model is Q_{MF}
  - Spoiler: TD(λ)
- Parameter → 4 (TD(λ) + ω)

Any Questions?

Short Break

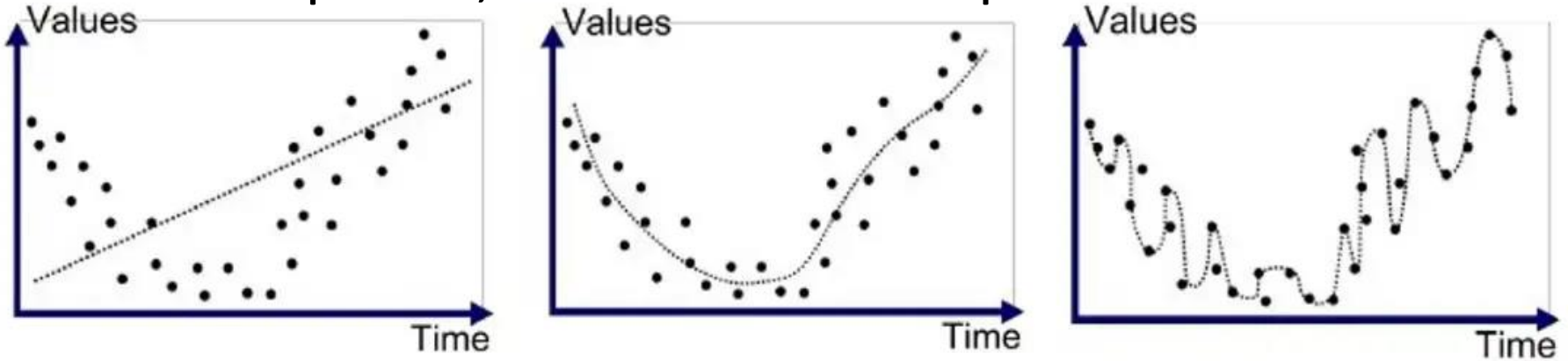
(If questions come up during the break, we discuss them after the break)

Model fitting and evaluation

- The model that describes the participant's data the best is the best model to describe human behavior in general?

- No, the models have different numbers of parameters. Models with more parameters could just be better because they are overfitting
- TD(0) = 2 params; TD(1) = 2 params; TD(λ) = 3 params;

Model-based = 1 param; Mixed-Model = 4 param



https://medium.com/analytics-vidhya/understanding-overfitting-and-underfitting-in-machine-learning-2a2f3577fb27

Model fitting and evaluation

- No, we have models that have different numbers of parameters. Models with more parameters could just be better because they are overfitting
- TD(0) = 2 params; TD(1) = 2 params; TD(λ) = 3 params;

Model-based = 1 param; Mixed-Model = 4 params

$$NLL(\boldsymbol{X}) = -\sum_{t=1}^{n} logp(a_t|\boldsymbol{X})$$

$$\boldsymbol{X}_{\text{MLE}} = arg\min_{x} NLL(\boldsymbol{X})$$

$$BIC = klogn + 2\boldsymbol{X}_{\text{MLE}}$$

$!NLL(X_{MLE})$

BIC = Bayesian Information Criterium -> lower values better

Results - TD Learning

Feel free to ask questions or start a discussion during the result slides !!

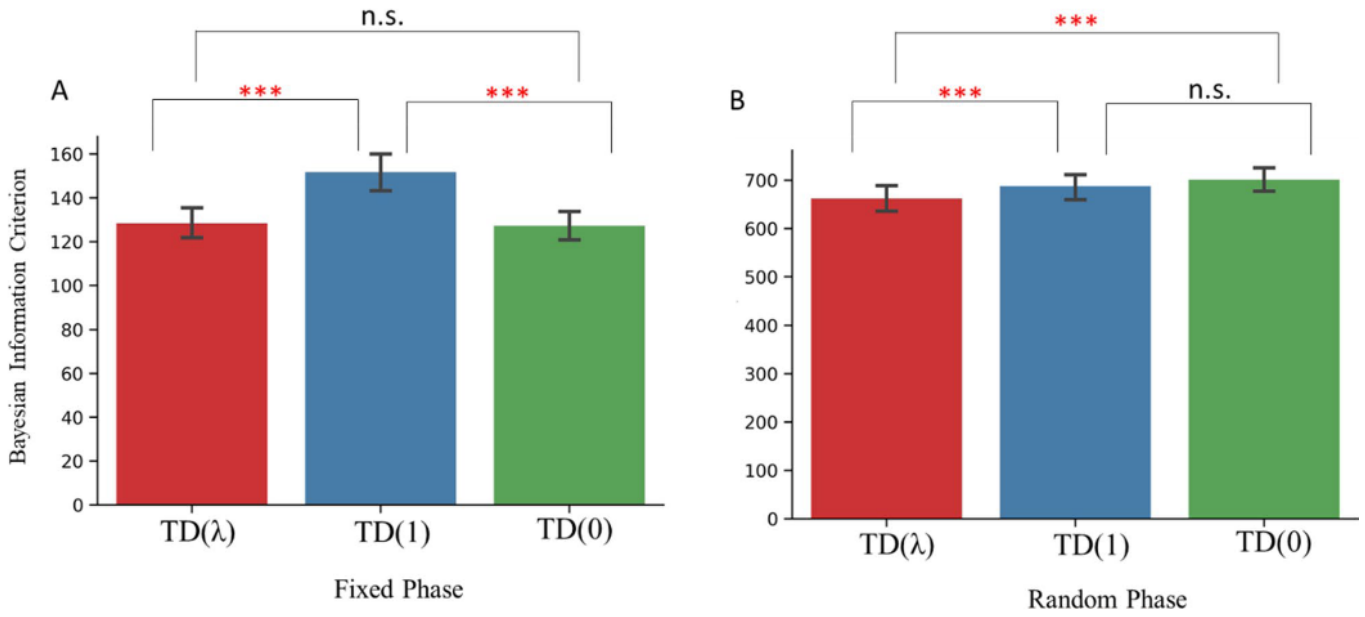


Figure 3. Model comparison in the Fixed (**A**) and the Random (**B**) phases. *BIC* Bayesian Information Criterion. *n.s.* not significant. ***$p < 0.001$.

Fixed Phase:

- TD(1) gets outperformed

Random Phase:

- TD(λ) performs best
- → λ modulates learning most realistic

Results - Model based vs. free

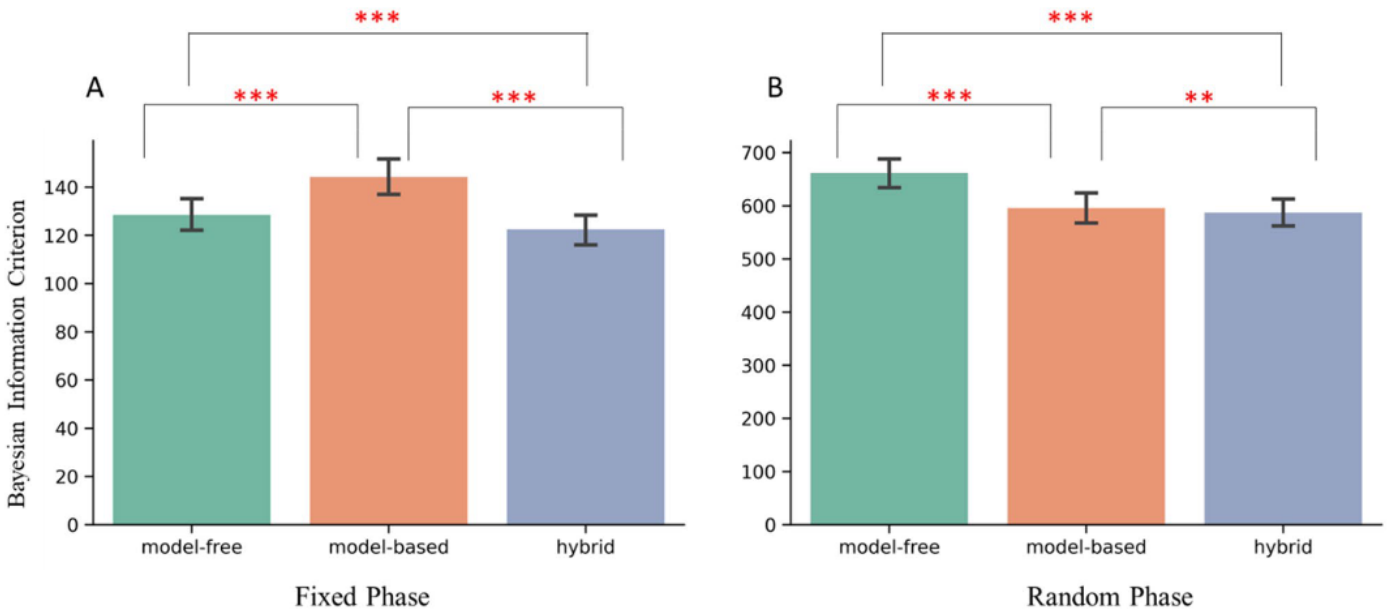


Figure 4. Model comparison in the Fixed (**A**) and the Random (**B**) phases. *BIC* Bayesian Information Criterion. **p < 0.01, ***p < 0.001.

Fixed Phase:

- No familiarity with environment

Random Phase:

- Model based represents the raising familiarity with environment
- Hybrid model → reliance on the Cognitive Map (participants)

Results – Navigation strategy

What can we learn from the difference between fixed and random phase?

H1: Familiarity to environment modulates between model free and based.

- ω: (High -> Model based; Low -> Model free)
- ω smaller in fixed phase

→ ω sign. smaller in fixed ($t(113) = -17.56$, $p < 0.001$)

Results – Navigation strategy

What can we learn from the difference between fixed and random phase?

H2: More exploration in random phase.

- θ : inverse temperature (High -> exploit; Low -> explore)
- θ decreases in random phase

→ θ sign. larger in fixed ($t(113) = -7.75$, $p < 0.001$)

Results – Navigation strategy

What can we learn from the difference between fixed and random phase?

H3: Strategies of good navigators differ between by requirements (phases).

- General: Cognitive Mappers are better Navigators
- → ω sign. correlated with Excessive Distance ($rs(114) < -0.51$, $ps < 0.001$)

- Fixed phase: Better Navigators use one strategy
  - High ω would lead to higher θ (exploit)
- → sign. positive correlation ($r(114) = 0.25$, $p = 0.007$)

- Random phase: Better Navigators vary strategy
  - High ω would lead to lower θ (explore)
- → sign. negative correlation ($r(114) = -0.35$, $p < 0.001$)

Q: *As a cognitive mapper, do you not - by nature - generally tend to explore (rather than exploit) more than route followers, since only by proper exploration, you can build a proper cognitive map of the environment?*
- Do Cognitive Mappers in the fixed task aim to build a map

Summary - Insights

- TD(λ): Memory update while learning is best represented by λ
- Hybrid model represents human navigation best
- Model-free and Model-based navigation depends on familiarity with environment
- Cognitive Mappers: navigation requirements modulate strategy
  - Setup is fixed → exploit
  - Environment is more random → explore

Summary - Methodology

- Individual navigation strategy:
  - (Previously) *Solution index* = preference Shortcuts or Familiar routes
  - ω is more detailed over time than the solution index
  - Deeper insights into learning and using Maps onto navigational strategy

- θ (exploration – exploitation)
  - As additional measurement for navigation strategy

- Methodology can be used further to gain insights into navigation strategy

Discussion Questions

- The article gives some arguments for what makes a "good navigator", in a discussion from nature vs nurture perspective, how do you think these factors would be considered? Also, if a similar study were conducted with children or adolescents, could we expect good results with a hybrid model as well? Or would this model be too complex for not fully developed individuals?
- With increasing randomness of the wayfinding, the use of the model-based system increases over the model-free systems. Can this process be described as a switch and how and where could this happen in the brain?
- Which other RL models can be tested in the context of this study? The researchers state that hybrid model shows the best fit to the behavioral data, however it might me just another (better) model-based method or (better) model-free approach.

Discussion – Our questions

- Do the RL models represent a wide enough variety of navigational constraints. Are possibilities missing?
  - Why no hybrid model like DynaQ?
- „People who completely rely on a model-based system are assumed to have a perfect cognitive map”(p.2) – Do you agree?
  - (this would be (at least partially) be the basis to conclude on omega)
- Cognitive Mappers are more efficient navigators. But solely model based performed worse than hybrid. What makes the additional use of model free learning so important.
- The Dual Solution task simplifies the navigation requirements. How do you think can the task be improved to gain further insights (For other participants, other models, other parameters)

Sources

- Dolan, R. J., & Dayan, P. (2013). Goals and habits in the brain. Neuron, 80(2), 312–325.
- He, Q., Liu, J.L., Eschapasse, L. *et al.* A comparison of reinforcement learning models of human spatial navigation. *Sci Rep* **12**, 13923 (2022). https://doi.org/10.1038/s41598-022-18245-1
- Sutton, R. S. & Barto, A. G. *Reinforcement Learning, Second Edition: An Introduction* (MIT Press, 2018).
- Image overfitting: https://medium.com/analytics-vidhya/understanding-overfitting-and-underfitting-in-machine-learning-2a2f3577fb27

Appendix

Initialize $Q(s, a)$ arbitrarily and $e(s, a) = 0$, for all s, a
Repeat (for each episode):
  Initialize s, a
  Repeat (for each step of episode):
    Take action a, observe r, s'
    Choose a' from s' using policy derived from Q (e.g., ϵ-greedy)
    $a^* \leftarrow \arg\max_b Q(s', b)$ (if a' ties for the max, then $a^* \leftarrow a'$)
    $\delta \leftarrow r + \gamma Q(s', a^*) - Q(s, a)$
    $e(s, a) \leftarrow e(s, a) + 1$
    For all s, a:
      $Q(s, a) \leftarrow Q(s, a) + \alpha\delta e(s, a)$
      If $a' = a^*$, then $e(s, a) \leftarrow \gamma\lambda e(s, a)$
                else $e(s, a) \leftarrow 0$
    $s \leftarrow s'; a \leftarrow a'$
  until s is terminal

- *Algorithm for Q-Value updating with eligibility trace*

Appendix – Some Critic

- Model fitting process
  - BIC penalizing of hybrid model (false formula)
  - Averaging Process over Model free learners is not described
- Unclear TD(1) implementation
- Multiple false formulations
  - like in Hypothesis 2: Higher exploration would lead to higher theta
- Gamma scale for Q(s+1, a+1) missing without explanation
- Fixed gamma value for model-based learning
- Added **after** presentation (to keep an overview about the authors response if someone writes a mail):
  - No description which size omega or lambda has, and how it develops over trials
  - Pearson coefficient not the right tool to measure correlation between theta and omega