Multi-intention Inverse Q-learning for Interpretable Behavior Representation

Hao Zhu Brice De La Crompe Gabriel Kalweit
Artur Schneider Maria Kalweit Ilka Diester Joschka Boedecker

Department of Computer Science
University of Freiburg

September 5, 2024

Outline

Introduction

Hierarchical inverse Q-learning

Experiments

Conclusion

Outline

Characterizing decision-making behavior

inverse reinforcement learning (IRL)

- consists in determining the underlying (intrinsic) reward function given expert demonstrations
- appears to be emerging as a valuable tool for constructing mathematical behavior models in behavioral neuroscience and cognitive science research

multi-intention IRL

- extends IRL from the single, fixed reward function to multiple, non-stationary reward functions
- considers that animal's goals can evolve over time due to, *e.g.*, fatigue, satiation, and curiosity

Related work

dynamical inverse reinforcement learning (DIRL) (Ashwood *et al.* [AJP22])

- extends maximum entropy IRL to non-stationary rewards
- achieved SOTA performance in animal behavior prediction
- parametrizes the animal's reward function as a smoothly time-varying linear combination of a small number of spatial reward maps with Gaussian random walk prior over weights

$$r_t(s) = \sum_{k=1}^{K} \alpha_{k,t} u_k(s)$$

  - $u_k \in \mathbf{R}^{|\mathcal{S}|}$: the kth reward map
  - $\alpha_{k,t} \in \mathbf{R}$: reward map mixing weight, where $\alpha_{k,t} = \alpha_{k,t-1} + \epsilon_k$ with $\epsilon_k \sim \mathcal{N}(0, \sigma_k^2)$
- allows the instantaneous reward function to vary **continuously** in time

▶ demands have emerged on IRL with **discrete** time-varying reward functions, especially after [ARS+22] suggesting that animals alternate between discrete strategies during decision-making

Outline

Inverse Q-learning

(Kalweit *et al.* [KHWB20])

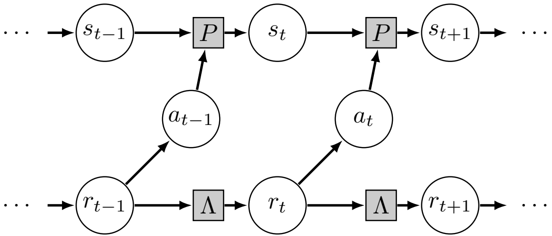
$$\begin{array}{ll} \text{maximize} & \mathbf{E}_{\xi \sim \mathcal{D}} \left[\log \mathbf{P}\left(\xi \mid \pi_r\right)\right] \\ \text{subject to} & \pi_r(s,a) = \exp\left(Q(s,a) - \log \sum \exp Q(s,\cdot)\right), \text{ for all } s \in \mathcal{S},\ a \in \mathcal{A} \\ & Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s,a,s') \max_{a' \in \mathcal{A}} Q(s',a'), \text{ for all } s \in \mathcal{S},\ a \in \mathcal{A} \end{array}$$

- optimization variable r: the unknown reward function
- problem data $\mathcal{D}$: the set of expert demonstrations with each trajectory $\xi \in \mathcal{D}$ defined as a sequence of state-action pairs: $\xi = \{(s_0, a_0), \ldots, (s_n, a_n)\}$
- Boltzmann policy constraint guarantees the IRL problem is tractable
- the transition probability P is not necessarily known
  - model-based: closed-form **inverse action-value iteration (IAVI)** via least squares
  - model-free: **inverse Q-learning (IQL)** via stochastic approximation

Graphical representation of expert's decision process

assumptions

- each expert demonstration is generated according to the Boltzmann optimal policy under one of the reward functions in the set $\mathcal{R} = \{r_1, \ldots, r_K\}$, with each corresponding to one specific intention
- the probability that one demonstration is generated under reward function $r \in \mathcal{R}$ is controlled by a Markov chain with initial state distribution Π and transition matrix Λ



Hierarchical inverse Q-learning (HIQL)

solving IRL problems on such decision network with parameters $\Theta = \{\Pi, \Lambda, \mathcal{R}\}$ consists in determining

- a set of reward functions
- the reward function index for each demonstration

consider the **expectation-maximization (EM)** approach, let $\eta = \{z_0, \ldots, z_n\}$ be the predicted sequence of reward function indexes for trajectory $\xi \in \mathcal{D}$, each iteration of the EM process can be written as an MLE problem:

$$\text{maximize} \quad J\left(\Theta^+ \mid \Theta\right) = \mathbf{E}_{\xi \sim \mathcal{D}, \eta}\left[\log \mathbf{P}\left(\xi, \eta \mid \Theta^+\right)\right]$$

- optimization variable: Θ^+
- problem data: $\mathcal{D}$ and Θ
- the predicted indexes η is marginalized out in the expectation

Hierarchical inverse Q-learning (HIQL)

solving the problem in page 9 is equivalent to solving a sequence of optimization problems:

$$\begin{array}{ll} \text{maximize (over } \Pi^+) & \mathbf{E}_{\xi\sim\mathcal{D}}\left[\sum_{i=1}^{K} \mathbf{P}(z_0 = i \mid \xi, \Theta) \log \Pi_i^+\right] \\ \text{subject to} & \Pi^+ \succeq 0,\ \mathbf{1}^T\Pi^+ = 1 \end{array}$$

$$\begin{array}{ll} \text{maximize (over } \Lambda^+) & \mathbf{E}_{\xi\sim\mathcal{D}}\left[\sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{t=1}^{n} \mathbf{P}(z_{t-1} = i, z_t = j \mid \xi, \Theta) \log \Lambda_{ij}^+\right] \\ \text{subject to} & \Lambda_{i:}^+ \succeq 0,\ \mathbf{1}^T\Lambda_{i:}^+ = 1, \quad i = 1, \ldots, K \end{array}$$

$$\begin{array}{ll} \text{maximize (over } r_i^+) & \mathbf{E}_{\xi\sim\mathcal{D}}\left[\sum_{t=0}^{n} \mathbf{P}(z_t = i \mid \xi, \Theta) \log \pi_{r_i^+}(s_t, a_t)\right] \\ \text{subject to} & \pi_{r_i^+}(s,a) = \exp\left(Q(s,a) - \log\sum\exp Q(s,\cdot)\right), \text{ for all } s \in \mathcal{S},\ a \in \mathcal{A} \\ & Q(s,a) = r_i^+(s,a) + \gamma\sum_{s'\in\mathcal{S}} P(s,a,s')\max_{a'\in\mathcal{A}} Q(s',a'), \text{ for all } s \in \mathcal{S},\ a \in \mathcal{A} \end{array}$$

- the Baum-Welch algorithm can be applied to obtain the posterior probabilities $\mathbf{P}(z_t = i \mid \xi, \Theta)$ and $\mathbf{P}(z_{t-1} = i, z_t = j \mid \xi, \Theta)$

Hierarchical inverse Q-learning (HIQL)

- the first two optimization problems about Π^+ and Λ^+ is maximized by

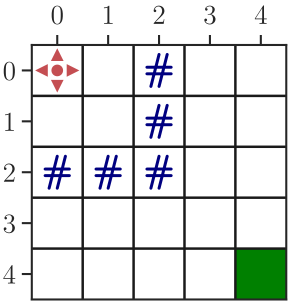
$$\Pi_i^+ = \mathop{\mathbf{E}}_{\xi \sim \mathcal{D}} [\mathbf{P}(z_0 = i \mid \xi, \Theta)], \quad i = 1, \dots, K$$

$$\Lambda_{ij}^+ = \frac{\mathbf{E}_{\xi \sim \mathcal{D}, t} [\mathbf{P}(z_{t-1} = i, z_t = j \mid \xi, \Theta)]}{\mathbf{E}_{\xi \sim \mathcal{D}, t} [\mathbf{P}(z_{t-1} = i \mid \xi, \Theta)]}, \quad i = 1, \dots, K, \quad j = 1, \dots, K$$

- the optimization problem about r_i^+ can be solved by first sampling a demonstration subset $\mathcal{D}'$ corresponding to r_i^+ w.r.t. $\mathbf{P}(z_t = i \mid \xi, \Theta)$, and then use the class of IQL algorithms to learn r_i^+ based on the sampled trajectories

Outline

Gridworld benchmark



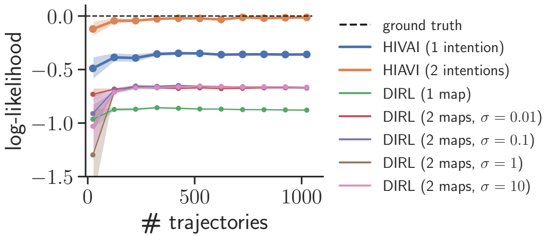
- $\mathcal{A} = \{$*left*, *right*, *up*, *down*, *stay*$\}$
- 10% probability to random state

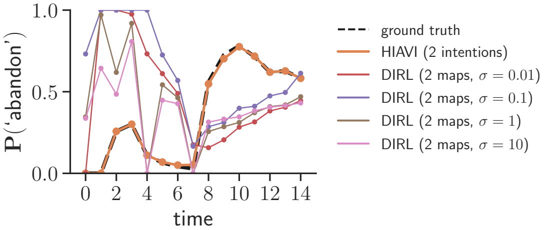
- π^{goal}: move towards $(4, 4)$
- π^{abandon}: move towards $(0, 0)$

---

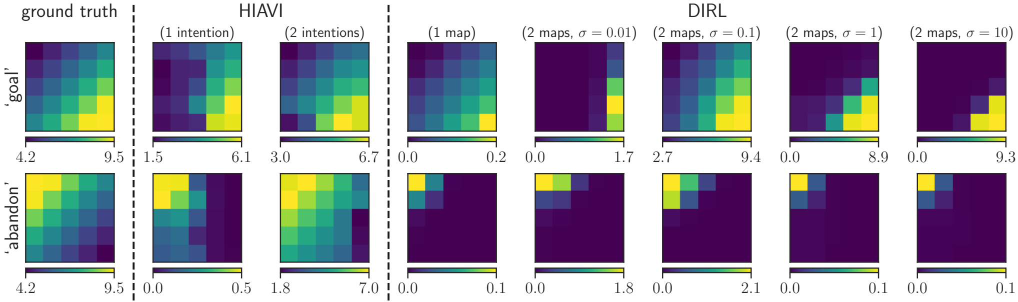
initialize $s := (0, 0)$, $\pi := \pi^{\text{goal}}$, $t := 0$.
repeat
  $a \sim \pi$.
  $s \sim P(s, a, \cdot)$.
  if s has barrier '#' **then**
    Switch to another policy (30%).
  else if $t = 8$ **then**
    $\pi := \pi^{\text{abandon}}$ (50%).
  end if
  $t := t + 1$.
until $(0, 0)$ or $(4, 4)$ is reached.

Gridworld benchmark



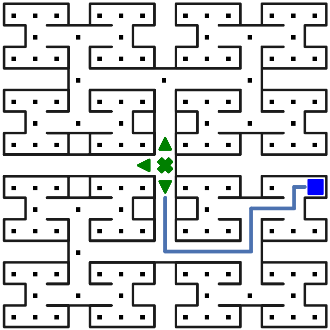


Gridworld benchmark



	HIAVI		DIRL				
				2 maps			
	1 intention	2 intentions	1 map	$\sigma = 0.01$	$\sigma = 0.1$	$\sigma = 1$	$\sigma = 10$
'goal'	13.96 ± 0.23	$\mathbf{5.58 \pm 0.47}$	47.10 ± 0.00	27.63 ± 6.55	2.97 ± 0.67	11.08 ± 3.56	37.97 ± 1.29
'abandon'	45.55 ± 0.07	$\mathbf{6.39 \pm 1.80}$	48.17 ± 0.00	45.90 ± 0.15	45.48 ± 0.49	48.15 ± 0.05	48.15 ± 0.06

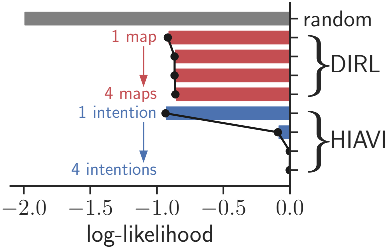
Real-world mice navigation benchmark

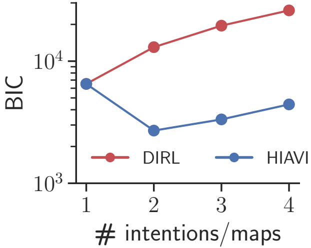


- action space: $\mathcal{A} = \{\textit{left}, \textit{right}, \textit{reverse}, \textit{stay}\}$
- $\mathbf{card}\,\mathcal{S} = 127$
- deterministic transition function P
- subjects: water restricted & unrestricted mice

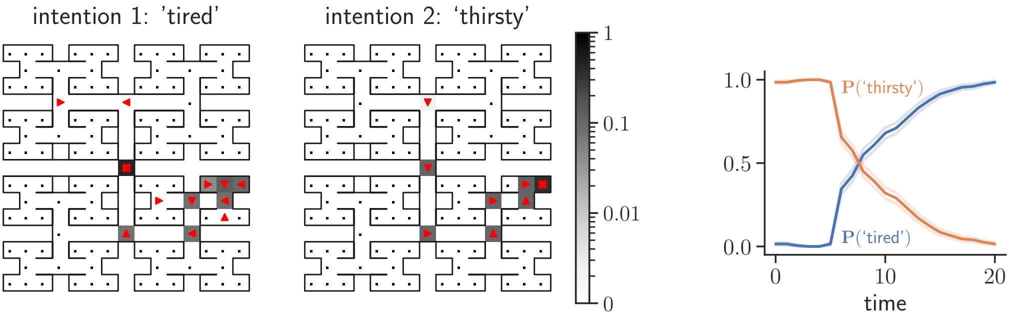
Real-world mice navigation benchmark

water restricted animals



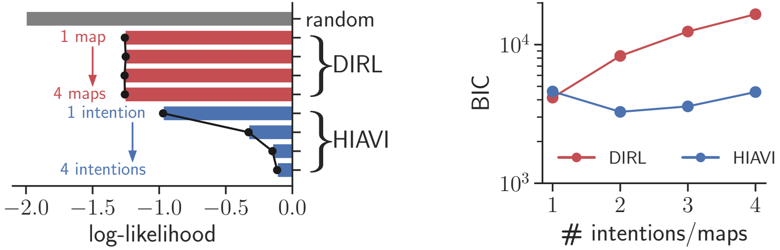


Real-world mice navigation benchmark

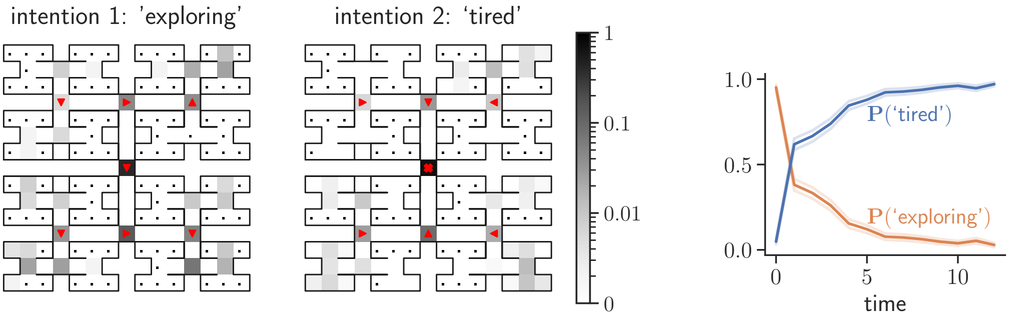


Real-world mice navigation benchmark

water unrestricted animals

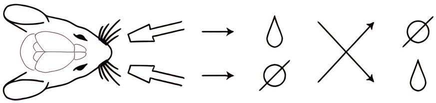


Real-world mice navigation benchmark



Application to mice reversal-learning behavior

dynamic two-armed bandit task



- deterministic reward (water) delivery
- performance-dependent reward switch

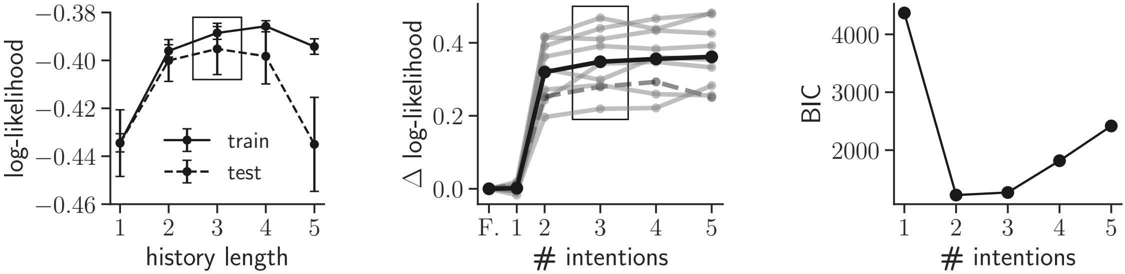
MDP formulation

- action space: $\mathcal{A} = \{\textit{left}, \textit{right}\}$
- state space:

$$s_t = (\varphi_{t-1}, \ldots, \varphi_{t-\ell_h}; a_{t-1}, \ldots, a_{t-\ell_h})$$

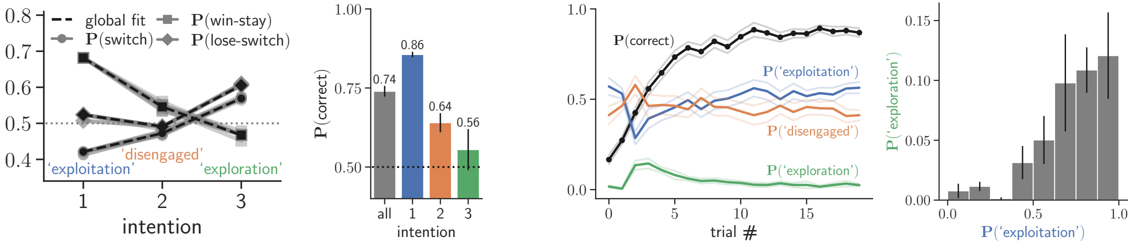
  - for all $s_t \in \mathcal{S}$
  - $\ell_h \in \mathbf{Z}_{++}$ — history length
  - $\varphi \in \{\textit{hit}, \textit{error}\}$ — history extrinsic reward
  - $a \in \mathcal{A}$ — history action
- unknown stochastic environment model P

Application to mice reversal-learning behavior



- F.: forgetting Q-learning model [BNLS22]

Application to mice reversal-learning behavior



Outline

Conclusion

the class of HIQL algorithms

- outperforms the SOTA on both synthesized and real-world datasets
- can produce interpretable behavior characteristics
- characterized typical exploration behavior of rodents during value-based decision-making

compared to the SOTA for characterizing animal behavior,

- the assumptions about the underlying intention transition dynamics in HIQL align better with those observed in real-world behavioral experiments

Reference

[AJP22] Zoe Ashwood, Aditi Jha, and Jonathan W Pillow.
Dynamic inverse reinforcement learning for characterizing animal behavior.
Advances in Neural Information Processing Systems, 35:29663–29676, 2022.

[ARS+22] Zoe C Ashwood, Nicholas A Roy, Iris R Stone, International Brain Laboratory, Anne E Urai, Anne K Churchland, Alexandre Pouget, and Jonathan W Pillow.
Mice alternate between discrete strategies during perceptual decision-making.
Nature Neuroscience, 25(2):201–212, 2022.

[BNLS22] Celia C Beron, Shay Q Neufeld, Scott W Linderman, and Bernardo L Sabatini.
Mice exhibit stochastic and efficient action switching during probabilistic decision making.
Proceedings of the National Academy of Sciences, 119(15):e2113961119, 2022.

[KHWB20] Gabriel Kalweit, Maria Huegle, Moritz Werling, and Joschka Boedecker.
Deep inverse Q-learning with constraints.
Advances in Neural Information Processing Systems, 33:14291–14302, 2020.