What Did You Think Would Happen?
Explaining Agent Behaviour Through Intended Outcomes

Herman Yau* (h.yau@surrey.ac.uk) Chris Russell(cmruss@amazon.com) Simon Hadfield (s.hadfield@surrey.ac.uk)

CVSSP Centre for Vision, Speech and Signal Processing

amazon | science

UNIVERSITY OF SURREY

1 - Overview

We proposed a novel type of explanation for value-based RL agents based around intended outcomes.

We proved that no post-hoc method can explain the intention of RL agents.

Our modifications to standard tabular-based RL methods provide these explanations, and we proved their consistencies with vanilla value-based RL.

We extended our method to deep neural networks.

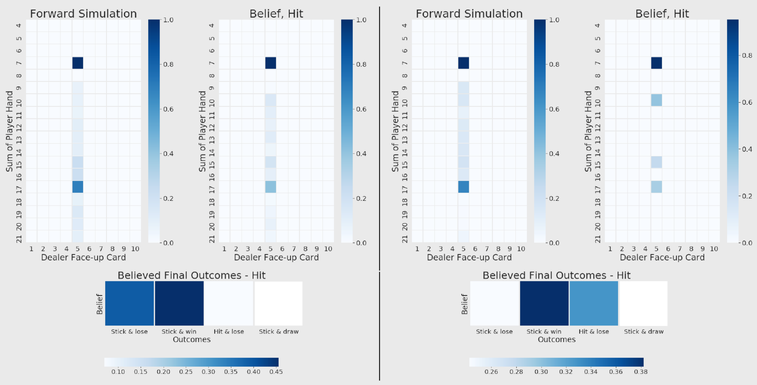
Code is available at github.com/hmhyau/rl-intention for custom OpenAI Gym environments.

2 - Motivation

Debugging: RL agents are notoriously hard to diagnose

Value Ambiguity: Unable to identify the trajectory the policy will follow from value alone

Training-testing misalignments: Insights are hard to identify after an agent is trained; colour indicates the likelihood of outcomes. See blackjack example below.



Good Agent: Behaviour vs Intention

Biased Agent: Behaviour vs Intention

3 - Value Function Decomposition

Aim: Train a map to project the agent's expected future.

We define **R** as the map of per state-action rewards, and **H** as the map of agent's expected future state visitations.

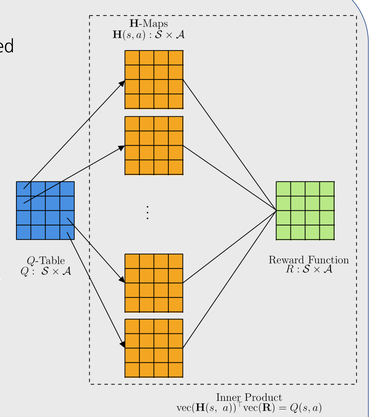
The Q-function is simply the inner product of the **H** and **R**. See figure on the right.

The proof of consistency is in the paper.

$$\mathrm{vec}(\mathbf{H}(s,a))^\top \mathrm{vec}(\mathbf{R}) = Q(s,a) \quad \forall a \in \mathcal{A}, s \in \mathcal{S}$$

We learn **H** during training time;

At test time **H** describes the agent's intended outcome for an action based on training experiences.



3 - Value Function Decomposition cont.

3.1 - Tabular Value-based Reinforcement Learning

H mimics updates of the Q-function.

We show Q-learning update here:

$$\mathbf{H}(s_t,a_t) \leftarrow \mathbf{H}(s_t,a_t) + \alpha\left(1_{s_t,a_t} + \gamma \mathbf{H}(s_{t+1}, \arg\max_{a\in\mathcal{A}} Q(s_{t+1},a) - \mathbf{H}(s_t,a_t)\right)$$

A similar extension for Monte-Carlo control can also be written.

$$\mathbf{H}(s_t,a_t) \leftarrow \mathbf{H}(s_t,a_t) + \alpha\left(\sum_{t=t'}^{T} \gamma^t 1_{s_t,a_t} - \mathbf{H}(s_t,a_t)\right)$$

3.2 - Application to DQN

We approximate the **H**-map by a deep neural network-based function approximator.

$$L_h(\theta_h) = \mathop{\mathbb{E}}_{(s_t,a_t,s_{t+1})}\left(1_{s_t,a_t} + \gamma \mathbf{H}\left(s_t, \arg\max_{a\in\mathcal{A}} Q(s_{t+1},a;\theta); \theta_h^-\right) - \mathbf{H}(s_t,a_t;\theta_h)\right)^2$$

This loses consistency guarantees, but performs well in practice.

4 - Results

4.1 - Blackjack

Goal: Get a player sum higher than the dealer's total.

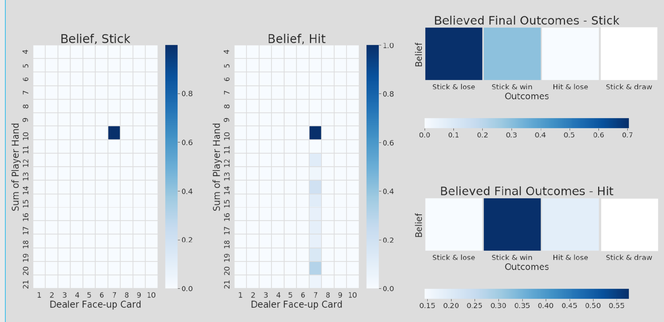
In this example the dealer initially shows a face-up card of 7 and the player shows a sum of 10.

We show the **H**-maps (*belief*) of the Q-learning agent. Each non-white point corresponds to a possible future the agent expects.

The peak at the starting state shows that the agent must pass through this state.

The second peak is present when player sum is 20 which indicates it is most probable to draw a card with a value of 10.

Together the **H**-map shows that hitting makes winning more likely in this case.



Expected future states and outcomes for the two available actions, stick and hit

4 - Results cont.

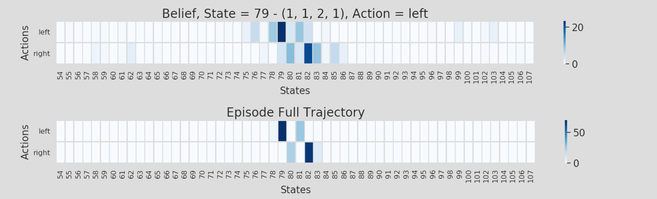
4.2 - Cartpole

Goal: Keep the pole of the cart upright on a frictionless track.

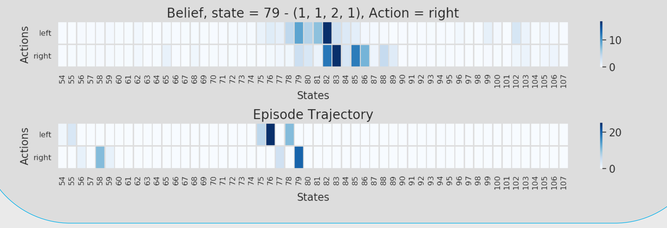
Continuous states in cartpole were quantised into a total of 162 states, consistency proof still applies. See the main paper for details.

We contrast the ***H**-map* (top) versus a *forward simulation* (bottom) of the tabular Q-learning agent.

The agent learnt to alternate between certain states to achieve equilibrium. The fuzzier **H**-maps reflects the uncertainty caused by the quantisation.



DQN agent shares the same behaviour as the tabular agent despite the loss of consistency guarantees.



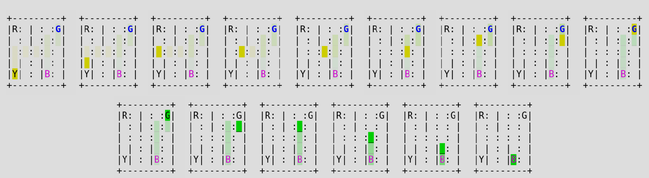
4.3 - Taxi

Goal: From a randomised start point, collect passenger from random location and take the person to a random destination.

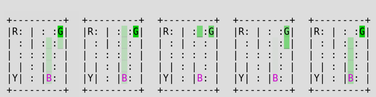
We present the visualised intentions of a trained Q-learning agent. Each map shows the corresponding timestep of a 15-step trajectory.

Agent starts at 'Y' intending to move to 'G' to pick up passenger, then move to 'B' to drop off. The intended route is followed faithfully.

The route becomes clearer as the effect of the discount factor becomes weaker.



Contrastive explanations can also be produced by the difference of **H**-maps from different actions. Agent intention remains unchanged.



From left to right: **H**-map of best and second best action, intention differences of first and second best actions, intersection of intentions