HOW TO EXPLAIN REINFORCEMENT LEARNING WITH SHAPLEY VALUES

University of Bath

Daniel Beechey, Thomas M. S. Smith, Özgür Şimşek

Bath Reinforcement Learning Lab

art-ai CDE

OVERVIEW

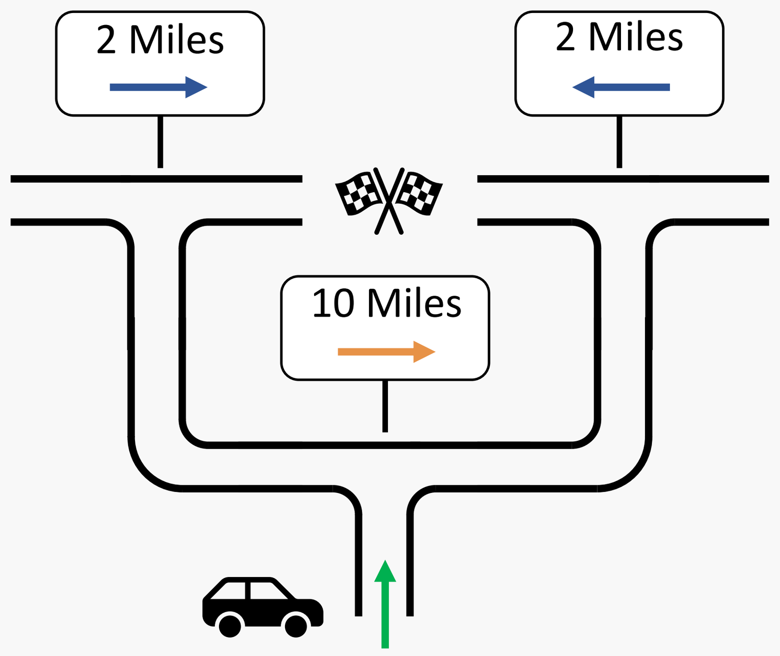
Despite its significant potential, such as controlling nuclear fusion reactors, uninterpretable agent behaviour hinders the deployment of reinforcement learning at scale.

We introduce **Shapley Values for Explaining Reinforcement Learning (SVERL)**, a mathematical framework for explaining agent-environment interactions.

WHAT ABOUT INTERACTIONS?

Certain features of an agent's observations influence different aspects of environment interactions: **policy, performance and performance prediction.**

Example: Autonomous vehicle using directions and distances (features) to navigate the shortest path to a destination.



COMPUTING FEATURE INFLUENCE

An example of the contribution assignment problem from cooperative game theory.

A **cooperative game** is a set of players $\mathcal{F}$ and a characteristic function $v: 2^{|\mathcal{F}|} \to \mathbb{R}$.

Contribution assignment problem: How to assign the contribution $\phi_i(v)$ of player i to the outcome of the game $(\mathcal{F}, v)$?

Shapley value: $$\phi_i(v) = \sum_{\mathcal{C} \subseteq \mathcal{F} \setminus \{i\}} \frac{|\mathcal{C}|!\,(|\mathcal{F}| - |\mathcal{C}| - 1)!}{|\mathcal{F}|!} [v(\mathcal{C} \cup \{i\}) - v(\mathcal{C})]$$

This unique solution satisfies four axioms specifying the fair distribution of credit across players.

CONTRIBUTION

SVERL is a mathematical framework for explaining agent-environment interactions using the influence of features on policy, performance and performance prediction.

In simple domains, SVERL produces meaningful explanations that match human intuition. In complex domains, the explanations reveal novel insight.

SHAPLEY VALUES FOR EXPLAINING REINFORCEMENT LEARNING (SVERL)

Three cooperative games played by the value of features $\mathcal{F}$ in state s whose outcomes are different aspects of agent-environment interactions.

1. EXPLAINING POLICY

Outcome: $\pi_s^a : 2^{|\mathcal{F}|} \to [0,1]$

The probability of selecting action a at state s when only the values of features $\mathcal{C}$ are known.

$$\pi_s^a(\mathcal{C}) \overset{\text{def}}{=} \mathbb{E}[\pi(S,a) \mid S_\mathcal{C} = s_\mathcal{C}] = \sum_{s' \in \mathcal{S}} p^\pi(s' \mid s_\mathcal{C})\,\pi(s',a).$$

The contribution of feature values to the probability of selecting action a in state s.

2. EXPLAINING PERFORMANCE

Outcome: $v_s^\pi : s^{|\mathcal{F}|} \to \mathbb{R}$

The expected return from state s when policy π only knows the values of features $\mathcal{C}$ at state s.

$$v_s^\pi(\mathcal{C}) \overset{\text{def}}{=} \mathbb{E}_\mu[G_t \mid S_t = s], \text{ where } \mu(s_t,a_t) = \begin{cases} \pi_{s_t}^{a_t}(\mathcal{C}) & \text{if } s_t = s, \\ \pi(s_t,a_t) & \text{otherwise.} \end{cases}$$

The contribution of feature values to performance from state s.

3. EXPLAINING PERFORMANCE PREDICTION

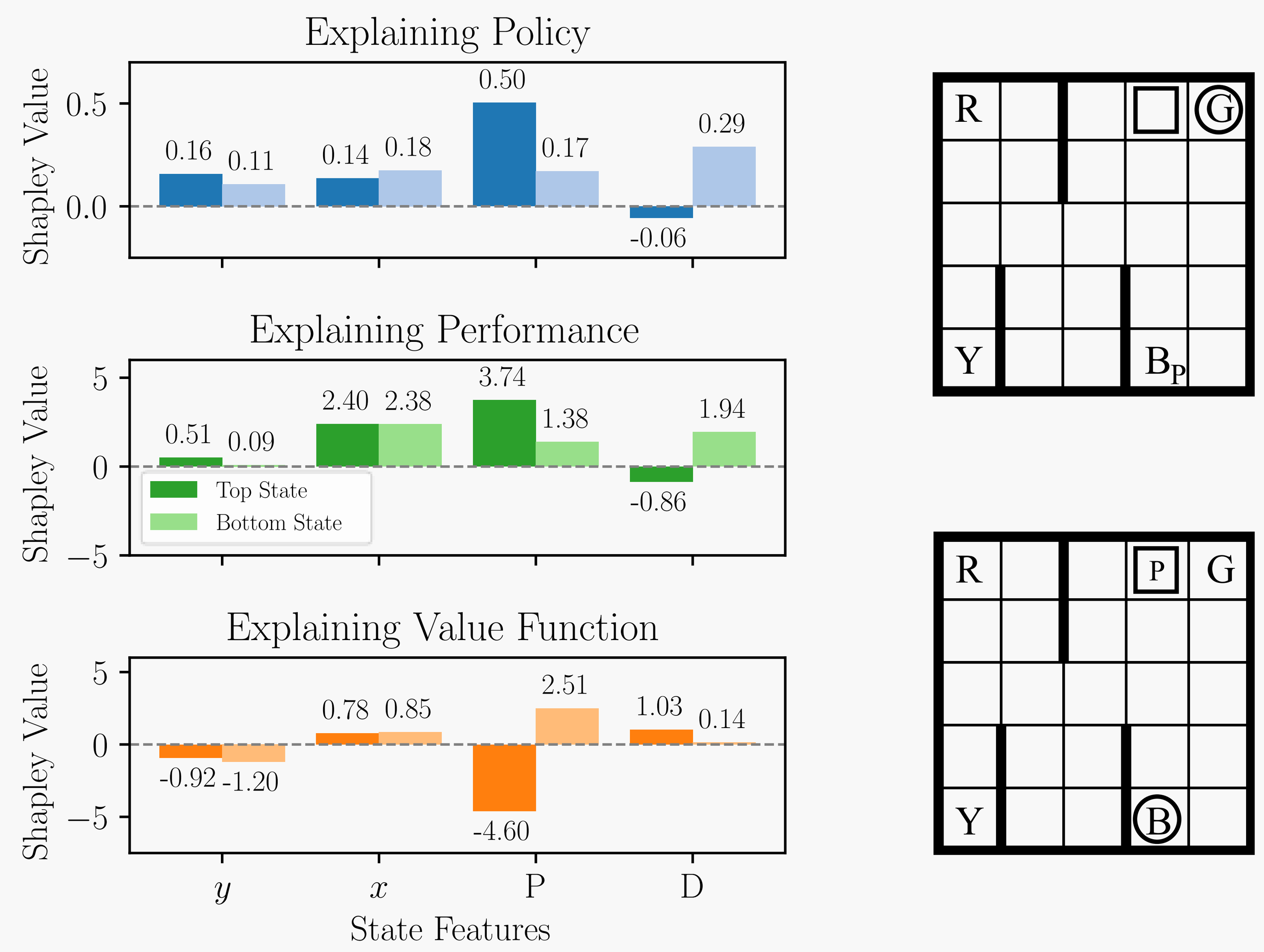
Outcome: $V_s^\pi : s^{|\mathcal{F}|} \to \mathbb{R}$

The expected return from observation $s_\mathcal{C}$ when following policy π.

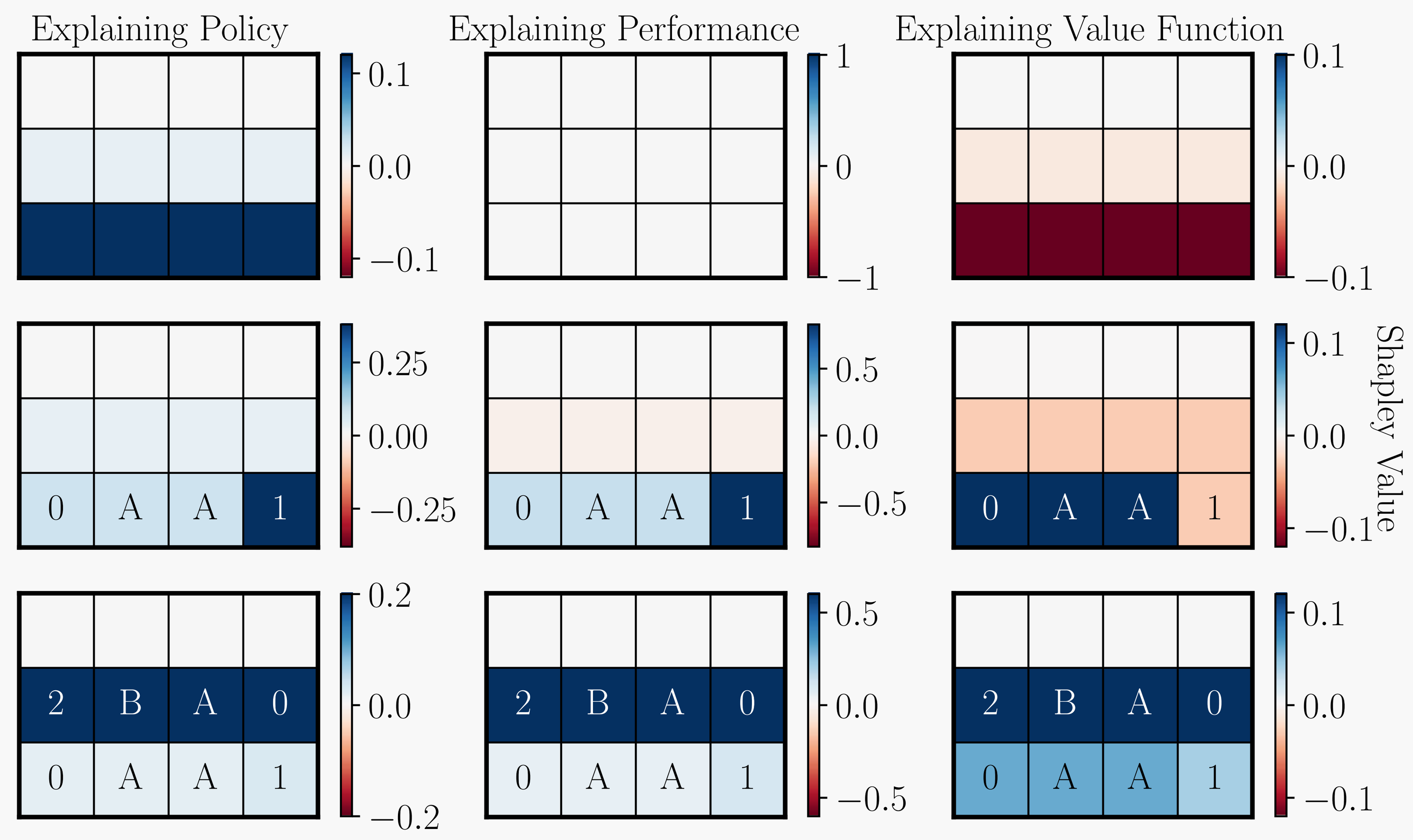
$$V_s^\pi(\mathcal{C}) \overset{\text{def}}{=} \mathbb{E}[v^\pi(S) \mid S_\mathcal{C} = s_\mathcal{C}] = \sum_{s' \in \mathcal{S}} p^\pi(s' \mid s_\mathcal{C})\,v^\pi(s').$$

The contribution of feature values to predicting performance from state s.

EXPLAINING TAXI



EXPLAINING MASTERMIND



OPEN QUESTIONS

1. How can SVERL be approximated in large domains?
2. How can the steady-state distribution $p^\pi(s)$ be efficiently approximated?
3. How can agent-environment interactions be explained for a continually learning agent?
4. How can combining explanation and behavioural models exploit shared structure to explain interactions as part of behaviour?

RESOURCES

[1] **Beechey, D.**, Smith, T.M. and Şimşek, Ö., 2023, July. Explaining reinforcement learning with Shapley values. In International Conference on Machine Learning (pp. 2003-2014). PMLR.

[2] Lloyd S Shapley. A value for n-person games. 1953.

Email: djeb20@bath.ac.uk

Website: https://djeb20.github.io/