Tracking and dissecting reward representations inferred from human adolescent behaviour in a risk-based decision-making task

Sean Spinney,[1,2] Guillaume Lajoie,[3,2] Irina Rish,[1,2]
Patricia Conrod,[4,5] and Maximilian Puelma Touzel[1,2]

[1] *Department of Computer Science, Université de Montréal*
[2] *Mila*
[3] *Department of Mathematics, Université de Montréal*
[4] *Department of Psychiatry, Faculty of Medicine, Université de Montréal*
[5] *CHU Ste Justine Hospital*

Quantitatively capturing the richness of behaviour when lacking a good decision-making model is a common challenge for both the behavioural sciences and for designing artificial agents that learn from experts. Maximum-entropy inverse reinforcement learning was developed in machine learning to address this challenge. Here, we apply it to model human adolescent behaviour in a risk-based decision-making task. We go beyond existing work by situating the observed behaviour in a feature space alongside different artificial agents that provide alternative, interpretable solutions to the task. In this comparative approach, we are able to glean more information about how the subjects might be using the features with which we describe their behaviour.

Keywords: human decision-making, reinforcement learning, BART task

I. INTRODUCTION

The behavioural sciences have long-faced a methodological challenge: quantifying and analyzing subject behaviour without explicit access to the decision-making process. A typical approach involves summarizing behaviour into statistics that can be used to test hypotheses generated *a priori* by the experimenter. However, a major limitation of such methods is that they strip away the context in which the observed behaviour was generated. In fact, there are often multiple ways of behaving that lead to the same statistics and it is often rather the sequence of decisions made within the experiment that are most revealing about what drives a decision-making process. Moreover, the *a priori* assumptions of what is important in a task that are baked into the structure of parametric models can make for highly biased inferences about what drives observed beahviour. The set of tools available for studying entire trajectories of decision-making without making strong assumptions on the behaviour remains lacking in psychology. A similar problem arises in artificial intelligence when developing algorithms that learn from observing experts, such as those for self-driving cars. For such cases, summary information is insufficient to produce human-like driving capabilities and methods have been developed that glean more from sequences of human decisions than merely their distribution. The intersection of these two fields is now an active research area involving behavioral science and machine learning, reinforcement learning in particular.

Maximum-entropy inverse reinforcement learning (hereon abbreviated as IRL) [10] offers an expressive, data-driven solution to identifying task features that drive observed behaviour. For a given set of statistics over a feature representation of the task, a parametrization

of the reward using these features, and a target objective computed from rewards (e.g. undiscounted return), IRL learns these parameters by matching the observed values of these statistics with a reward-driven agent while being otherwise as unstructured (*i.e.* as unbiased) as possible in how features lead to rewards. The reward representations learned can be independent of details of the agent model needed to generate behaviour. IRL provides a reward representation built on these set of features. When the latter sufficiently covers the features used by the observed agent, this representation can be a highly informative lens into the behaviour.

Here, we present our methodology for revealing and analyzing how a feature-based representation of a task drives a decision-maker's behaviour and to use it to derive scientific insight about how this representation changes over developmental timescales. We (1) infer from the observed behaviour a reward function, ie. a mapping from task states to subjective reward using IRL, (2) situate this representation among those inferred from the behaviour of some interpretable artificial agents, (3) verify the features that structure the reward representation, and (4) correlate this reward representation with other measurements to quantify their reward sensitivity. We confirm that the method matches the visitation statistics across these distinct agents. We find that the representation learned from human observations closely follows what is recovered from observing Q-learning behaviour that seeks to maximize expected returns. Humans perform suboptimally on the task, however, and in fact are risk-averse.

We then use the structure of the model to decompose the reward function into its feature and weight components in order to explain its overall shape. Only a single feature dominates and we show that both the sharp and broad components that the reward function empirically decomposes into are inherited from those same components in the save statistics over the state space. Armed with an understanding of this inferred reward function, we go to characterize the extended dataset that spans multiple years in order to track the development of the reward representation. We find the statistics used to train the model are unchanged across years, leading to similar feature weights and resulting performance. This is despite previous work suggesting that risk-taking behavior peaks during mid-adolescence [4–6]. Finally, we evaluate response times as a proxy for cognitive effort and correlate them with the recovered reward function demonstrating that they decouple over time. Together, these results demonstrates how IRL, augmented with generic methods for decomposing the resulting reward function, can be used to identify drivers of decision-making sufficient to explain the observed behaviour.

The inverse problem of behaviour and how IRL solves it

The well-established subjective nature of reward in human decision-making implies that quantifying rewards through extrinsic means (e.g. nutrient content), even in well-controlled experimental settings, is not likely to accurately reflect the intrinsic reward on which the subject bases their decisions. Cognitive biases, approximate inferences, and motivation all contribute to this discrepancy. Despite the unobserved nature of intrinsic reward, what is observed is the behaviour produced on its basis. Inverse reinforcement learning is a procedure to estimate an approximation to the intrinsic reward function, to the degree possible given the observed behaviour and under clear and minimal assumptions about what the agent is focussing on.

Imitation learning [1] uses state features and assumes a model for the reward function to

learn from observed behaviour e.g. of an expert. Typically, reward models are formed from a set of feature functions $\{\phi_j(\boldsymbol{s})\}_{j=1}^{F}$ acting on the task state $\boldsymbol{s} = (s, a)$ as the state-action tuple. A reward model based on a linear combination of these features is then

$$R(\boldsymbol{s}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{s}) \ ,$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_F)$ is the vector of feature weights and $\boldsymbol{\phi}(\boldsymbol{s}) = (\phi_1(\boldsymbol{s}), \ldots, \phi_F(\boldsymbol{s}))$ is the vector of feature functions that together map the state into a feature space. An agent policy then provides actions given this feature-based reward and corresponding behaviour can be compared with the actual behaviour. The similarity in this comparison then forms the basis for estimating the parameters of the approximate reward function that uses these features (see next section for details).

With a good fit to the actual behaviour, it is then instructive to inspect the learned reward function and the policy derived from it. For example, in this case of a linear feature approximation that matches the first-order feature statistics (the typical choice in existing work on MaxEnt IRL), the relative importance of the features $\boldsymbol{\phi}$ used in the decision-making can be assessed via their contribution to the recovered reward function, e.g. as a function of state.

The IRL algorithm estimates the parameters of this reward model by learning a policy which matches the actions taken and states visited by the expert. A trial of the task is a terminating trajectory of task states, i.e. a variable-length sequence of action-state tuples describing the actions taken and states visited during a single episode. In a sequence of trials indexed by $k = 1, \ldots, K$, let ζ_k denote the trajectory for episode k which ends after M_k steps so that $\zeta_k = \{\boldsymbol{s}_i = (s_i, a_i)\}_{i=0}^{M_k}$. With a slight abuse of notation, the total estimated reward for a trajectory is then denoted

$$R(\zeta) = \sum_{\boldsymbol{s}_i \in \zeta} \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{s}_i) \ . \tag{1}$$

This reward function on trajectories forms the basis for placing a value on the observed task trajectories. Assuming for the moment that the expert is acting optimally with respect to this function, inferring the correct weights from observed trajectories is ill-posed since different behaviours can lead to the same rewards. Taking the trivial case where all rewards are 0, then every policy will be optimal and there is no single identifiable set of optimal weights due to the degeneracy of the resulting solution. It is not possible to directly estimate the reward function due to this ambiguity. MaxEnt IRL never encounters this problem since it matches feature expectations instead (with convergence guarantees [1]), and fills in the rest with the criteria of being as unbiased as possible. This shifts the ambiguity to the feature matching, as different policies can lead to the same feature counts. Fortunately, the MaxEnt IRL algorithm mitigates the problem of identifiability by selecting the distribution over behaviours that maximizes entropy (i.e. minimizes additional assumptions) beyond matching some feature statistics. The Boltzmann distribution that results from this constrained optimization exponentially prefers actions with greater rewards, and its parameters can be obtained using relatively simple learning dynamics. While MaxEnt IRL was developed in machine learning, it is readily applicable to human behaviour on account of its general formulation.

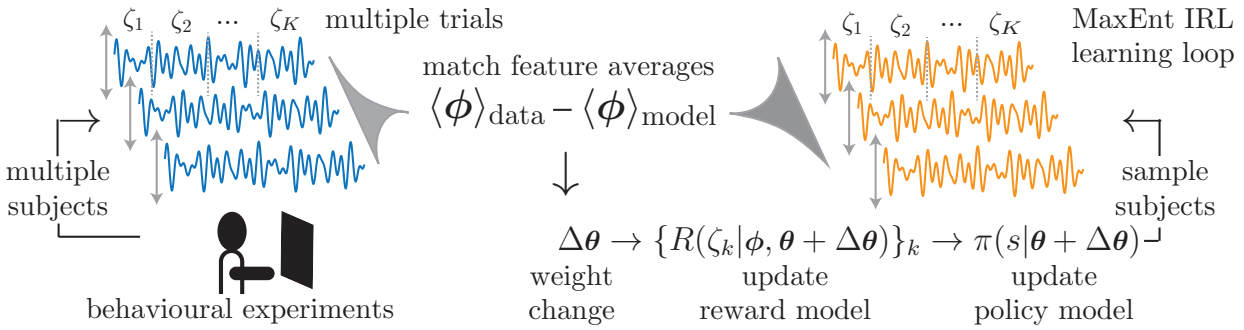

Figure 1. MaxEnt IRL applied to human decision-making. A set of behavioural (e.g. decision-making task) data is collected from many subjects, each doing many trials. The feature expectations, $\langle\boldsymbol{\phi}\rangle_{\text{data}}$, over the data are computed. The MaxEnt IRL learning loop then proceeds from the initialization of the model's feature weights, $\boldsymbol{\theta}$. These are used to compute the rewards accumulated over trajectories, from which a policy is formed. A model dataset of behaviour is then generated by sampling trials and behaving according to this policy. The model's feature expectations, $\langle\boldsymbol{\phi}\rangle_{\text{model}}$, are computed over this sample and compared with $\langle\boldsymbol{\phi}\rangle_{\text{data}}$ to determine the update applied to the feature weights. This procedure is repeated until the weights converge.

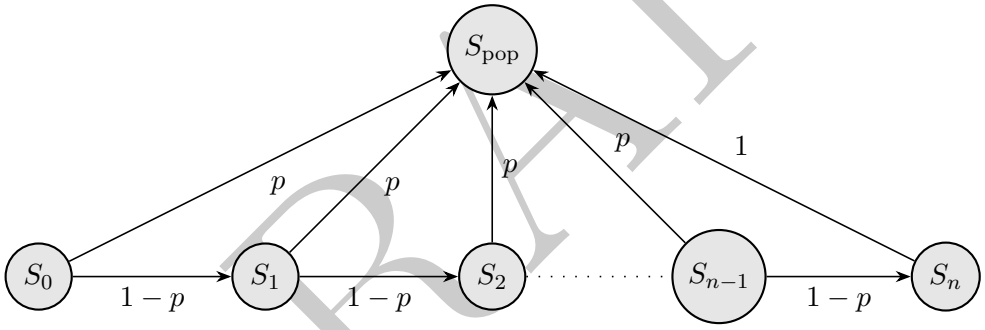

Figure 2. A single trial of the BART task. The pop probability for S_i, $i = 1, \ldots, n-1$, is $p = 1/(n-1)$.

II. METHODS

We first describe the task on which the subject data that we analyze was obtained, then go on to explain the implementation of the inverse reinforcement learning algorithm, followed by the steps and motivations taken to ensure reliable weight estimates. Lastly, we include a short description of the agents used to compare against the human results.

A. Behavioural experiment: BART description

The Balloon Analogue Risk Task (BART) is an episodic decision-making task, where the objective is to obtain the highest score over a fixed number of $K = 30$ trials (episodes). The mechanics of the discrete time task are as follows: a balloon (initially deflated) appears on a screen, along with a total score, and current trial score indicator (both initialized to 0). So long as the balloon has not yet popped, the agent has two possible actions at each time step; she can either pump a fixed amount of air into the balloon for a \$ 10 reward that adds

to her current trial's score, or save the points accumulated so far and move onto the next balloon. After every pump, the probability of the balloon popping increases and if it ever pops the agent loses her accumulated score on that trial. Subjects are instructed to try to obtain the highest score on every trial, and they are told that the larger the balloon is the greater the chance of popping.

More formally, we can describe the BART as a finite Markov Decision Process (MDP) where the state, S, is the number of pumps so far, denoted S_i with $i = 0, 1, ..., n = 128$ indexing the state space. Pump and save are the two possible values for the action, a. There are two terminal states, $S_{\text{save}}, S_{\text{pop}}$ serving as save and pop states respectively. If an agent is at the $(n-1)^{th}$ state and pumps, it will pop. The state at which the balloon pops on any given trial is drawn from a uniform distribution over the index. Then the probability of popping conditioned on reaching the i^{th} state is just the probability renormalized over the remaining states:

$$P(S_{\text{pop}}|S = S_i, a = \text{pump}) = \frac{1}{n+1-i} .$$

The balloon either pops or not after each pump so

$$P(S_{i+1}|S = S_i, a = \text{pump}) = 1 - P(S_{\text{pop}}|S = S_i, a = \text{pump}) = \frac{n-i}{n+1-i} .$$

The rewards obtained by the agent playing the game are

$$R(S_i, a = \text{pump}) = \begin{cases} -r_0(i-1), & \text{if popped} \\ +r_0, & \text{otherwise} , \end{cases}$$

where $r_0 = 10$ is the reward per pump. In order to respect the conditions of a Markov Process, the transition dynamics depend only on the previous state such that:

$$P(S_{i+1}|S_0, S_1, S_2, ..., S_i, a = \text{pump}) = P(S_{i+1}|S_i, a = \text{pump}) .$$

B. Maximum Entropy IRL

Matching feature expectations does not however guarantee that the recovered policy is unique. In fact, many policies can be optimal given a reward function, and they may even lead to the same feature counts [10]. The solution offered by the maximum entropy formulation to resolve this ambiguity is to use a distribution which assigns equal probabilities to policies with the same return, while exponentially preferring policies which obtain greater returns. Under this model, the distribution over trajectories becomes:

$$P(\zeta_k|\theta, T) = \frac{1}{Z(\theta)} e^{\sum_{\boldsymbol{s}_i \in \zeta_k} \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{s}_i)}$$

where T is the state transition probability matrix and Z is the normalization factor. Maximizing the entropy over trajectories amounts to maximizing the likelihood of the observed trajectories. As agents interact with the environment (e.g. in a decision-making task), they gain information over consecutive trials creating a dependency on past states. However,

if we encode information about past trials in the features, and assume now that the state features are independent across trials, we can write the log likelihood as [7]:

$$\ell(\boldsymbol{\theta}) = \sum_{k=1}^{K} \log P(\zeta_k|\boldsymbol{\theta}, T)$$

The gradient of this function with respect to the reward weights $\boldsymbol{\theta}$, is the difference between empirical feature counts, $\widetilde{\boldsymbol{\phi}} = \frac{1}{K}\sum_{k=1}^{K}\sum_{\boldsymbol{s}_i\in\zeta_k}\boldsymbol{\phi}(\boldsymbol{s}_i)$ and the learner's expected feature counts:

$$\nabla\ell(\theta) = \widetilde{\boldsymbol{\phi}} - \sum_{s_i} D(s_i, \boldsymbol{\theta})\boldsymbol{\phi}(s_i)$$

where $D(s_i, \boldsymbol{\theta})$ are the state visitation frequencies. If instead the feature vector is not fixed across trials, as is the case for our presentation of the BART, the feature vector becomes $\boldsymbol{\phi}^k_{s_i}$, and similarly $\tilde{\boldsymbol{\phi}}^k$ becomes the feature count for a single trajectory. We use a stochastic gradient descent implementation of the above gradient where the weights are updated after every trajectory. For example, after the k^{th} trajectory the reward weight vector $\boldsymbol{\theta}_k$ are updated using the gradient:

$$\nabla\ell^k_{SGD}(\theta_k) = \widetilde{\boldsymbol{\phi}}^k - \sum_{i=1}^{M_k} D(s_i)\boldsymbol{\phi}^k(\boldsymbol{s}_i) \tag{2}$$

where the feature matrix $\boldsymbol{\phi}^k$ is observed from the expert data. Note that $\boldsymbol{\phi}^k(S_{\text{pop}}, S_{\text{save}}) = 0$, $\forall k$. Thus, the weight dynamics are $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta\nabla\ell^k_{SGD}(\theta_k)$ for learning rate η. The state visitation frequencies $D(s_i)$ are generated from the learned policy derived using value iteration on the states and their corresponding estimated reward. As our estimate of the reward $R(\zeta)$ improves, so will our policy, which draws its learning signal from matching feature counts. This in turn should lead to matching behaviour on the task.

The log likelihood of the observed trajectories increases with the number of epochs, however the recovered policy generates matching state visitation frequencies after only a few epochs. This means that the weights are updated proportionally, such that the policy derived during the IRL step is the same on the next epoch. The stopping criterion was when the difference between the sum of subsequent normalized weights were smaller than an arbitrary value $\epsilon = 0.001$, and if the observed and estimated state visitation frequencies in (a) matched by eye.

C. Comparing learning agents

Validation of the IRL algorithm in uncovering characteristic learning styles is done by comparing different agents on the task. We select the following baseline algorithms to compare against human data:

1. random: makes a random choice at every pump

2. alwayspump: always selects the pump action

3. optimal: pumps until the 64^{th} pump then saves

4. Q-learning algorithm [8]

For a more controlled environment, the states which pop on any given trial are preselected such that identical policies lead to the same trajectories. The pop states are uniformly drawn (with replacement) over the set of possible states leading to an average pop state of 64.

III. RESULTS

We apply the algorithm described in the previous section to a BART task dataset of 138 adolescents (mean age = 14.92 years), obtained in the context of a larger study on early onset adolescent substance use[3]. As a baseline, we employ features used in a previous work [7] (see table I).

With the addition of an initial learning rate η_0 and decay schedule, the weight dynamics given by eq. (2) gave a wide range of convergence rates (not shown). To align the learning speeds of each feature, we gave each feature its own initial learning rate collected in the vector $\boldsymbol{\eta}_0 = (10^{-4}, 0.1, 0.1, 0.1, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4})$. With this modification, all results shown were obtained from weights that converged under the criteria of converging normalized weights. The state visitation frequencies are shown in **??** along with those of the data. The correspondence is excellent as expected since this is the target of the learning rule.

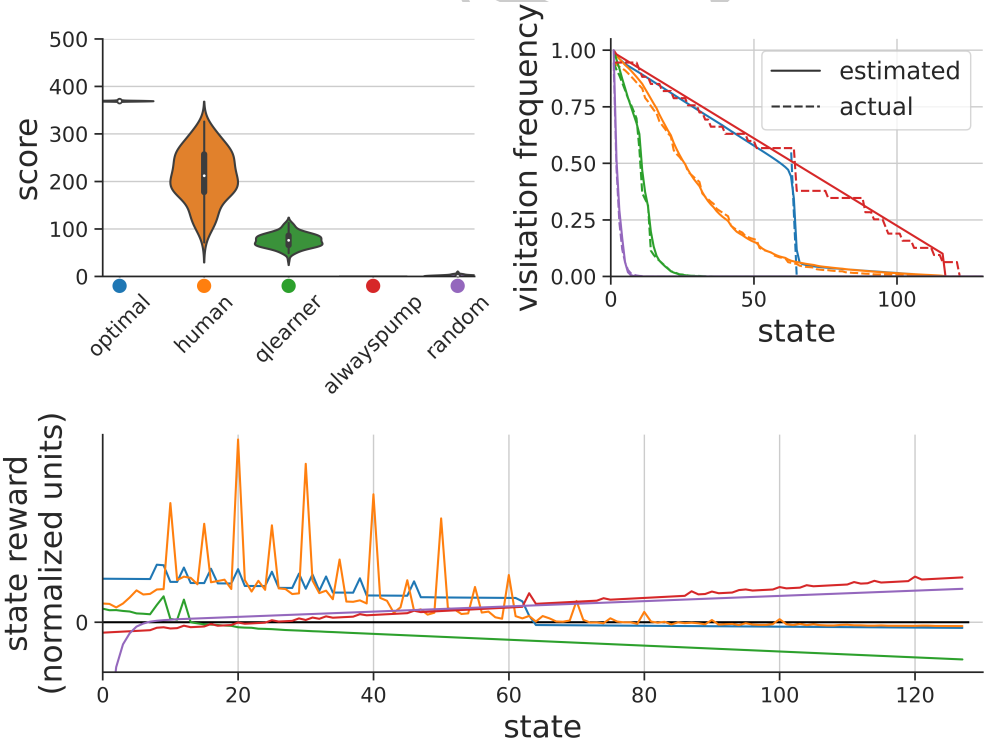

Figure 3. **Top left**: actual score obtained after 30 trials; **Top right**: actual versus estimated state visitation frequency; **Bottom**: estimated reward functions over states for each agent.

In order to benchmark the method, we implemented other artificial agents for the task and produced datasets of their behaviour. We applied the same IRL algorithm to obtain

a set of feature weights for each artificial agent, as was done with the observed data and using the same number of experts and trials; 138 and 30 respectively. The learned weights for the observed data and these artificial agents are shown in fig. S1. We normalize the weights across features for visualization only. We point out that there are many features with weights near 0 for all agents indicating the task can be solved using a limited number of important features, which is what is recovered from the optimal agent.

Since some features contribute for every state visited while others are only sparsely populated along the trajectory, the weights by themselves do not reflect the contribution of a feature to the total reward in a trajectory. Instead, we collect terms in eq. (1) by state, and plot their sums as a function of state (see fig. 3: B,C). Of note is that the most rewarding state estimated for the human matches the average save state computed from the observed data ($S_{avg} = 31.41$). Also, there is a similarity in this state-dependent reward around the average save state for the *qlearner* and humans. This is better seen by rescaling the x axis for each learner by their average save state (8.89, and 31.41, respectively). Meanwhile, the random agent's recovered reward function is very low across states; suggesting that the random trajectories aren't informative to contributing to finding a good policy from observations. The *alwayspump* agent leads to the recovered reward function negatively weighing early stages in the task, and valuing pumping more often. Finally, the *optimal* agent's recovered reward function is similar to the human's distribution, which would entail that the learning signal obtained from the human's data is informative and clear enough to approximate the reward function learned from an optimal agent.

Whereas fig. 3 plotted different learning agents against one another, fig. 4 focuses on the human results and reveals how the reward distribution in fig. 3B) is constructed from the observed features and the weights recovered by the IRL algorithm. Referring to Equation eq. (1), the result of the dot product between the reward weights $\boldsymbol{\theta}$ (fig. 4(a)) and the state features $\boldsymbol{\phi}(s_i)$ (fig. 4(b)) is shown in fig. 4(c). Furthermore, the contribution of each feature in the dot product is broken down.

The third weight, F3, dominates the reward landscape which can be explained by considering the magnitude of the weight and the distribution of the corresponding feature. The spikes in the reward distribution for F3 are aligned with states that are saved on very frequently (fig. 5(a)). These exhibit a frequency of 5, also clearly visible in the histograms of save states. This salient feature of the behaviour expressed continuously across the set of subjects suggests humans use a degree of time/state abstraction in the task: the creation of mental checkpoints that may ease the burden of estimating risk when deciding whether to continue pumping or save.

This last observation can be summarized in the hypothesis that a consequence of this behaviour is shorter reaction times on states between checkpoints.

DISCUSSION

The behaviour generated by the IRL algorithm's learned policy allows us to deconstruct how the task interacts with the agent, represented as features, in the decision-making process. If the true set of task properties can be reliably broken down into features, then a comprehensive picture can be drawn and different decision-making strategies can be compared. This is what the problem-solving aspect of cognitive psychology hopes to uncover: reliable descriptions of what drives decision-making which are flexible and adaptable to different environments, *i.e.* a modelling approach which generalizes well. These points have

been addressed by our approach in that we have shown that the MaxEnt IRL algorithm can be used to describe quantitatively, and qualitatively different decision-making approaches.

For the case of evaluating risky decision-making, there are good grounds for believing that reward processing plays a direct role in the variation in human and animal behaviour. For instance, there is growing evidence to suggest that valence processing develops asymmetrically through adolescence [2, 9]. This coincides with the neurobiological maturation gap observed during that time, and it potentially leaves developing humans vulnerable to dangerous outcomes (*e.g.* driving under the influence of alcohol). It then seems reasonable to assume that the space of internal reward representations may be more suited to study optimal and pathological decision-making, in the hopes of creating more powerful discriminative and descriptive behavioural analysis tools.

Future work could study the result of including features that characterize not the task, but psychological markers of the agent at the time of decision-making. Using the MaxEnt IRL algorithm, it would be possible to extract their relative contribution to how value is attributed to certain states. There needs to be careful examination of the efficacy of adding additional features. However, the methodology described in this work combined with an appropriate model selection criteria (*e.g.* information criterion) represents a unique opportunity for studying behaviour.

ACKNOWLEDGMENTS

We acknowledge Dr. Anqi Liu for helping clarify details of the algorithm used in [7]. This study was externally funded by an operating grant from Canadian Institutes of Health Research (Grant number: 126053; PC), consequently the protocol has undergone peer-review. The authors are partially supported by a fundamental research projects IVADO grant (ref: PRF-2019-4275683763). The study reference number for ethical approval is 3678.

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi:10.1145/1015330.1015430. URL https://doi.org/10.1145/1015330.1015430.

[2] Mariam Arain, Maliha Haque, Lina Johal, Puja Mathur, Wynand Nel, Afsha Rais, Ranbir Sandhu, and Sushil Sharma. Maturation of the adolescent brain. *Neuropsychiatric disease and treatment*, 9:449, 2013.

[3] Josiane Bourque, Travis E Baker, Alain Dagher, Alan C Evans, Hugh Garavan, Marco Leyton, Jean R Séguin, Robert Pihl, and Patricia J Conrod. Effects of delaying binge drinking on adolescent brain development: a longitudinal neuroimaging study. *BMC psychiatry*, 16(1): 445, 2016.

[4] Barbara R Braams, Anna CK van Duijvenvoorde, Jiska S Peper, and Eveline A Crone. Longitudinal changes in adolescent risk-taking: a comprehensive study of neural responses to rewards, pubertal development, and risk-taking behavior. *Journal of Neuroscience*, 35(18): 7226–7238, 2015.

[5] Stephanie Burnett, Nadège Bault, Giorgio Coricelli, and Sarah-Jayne Blakemore. Adolescents'

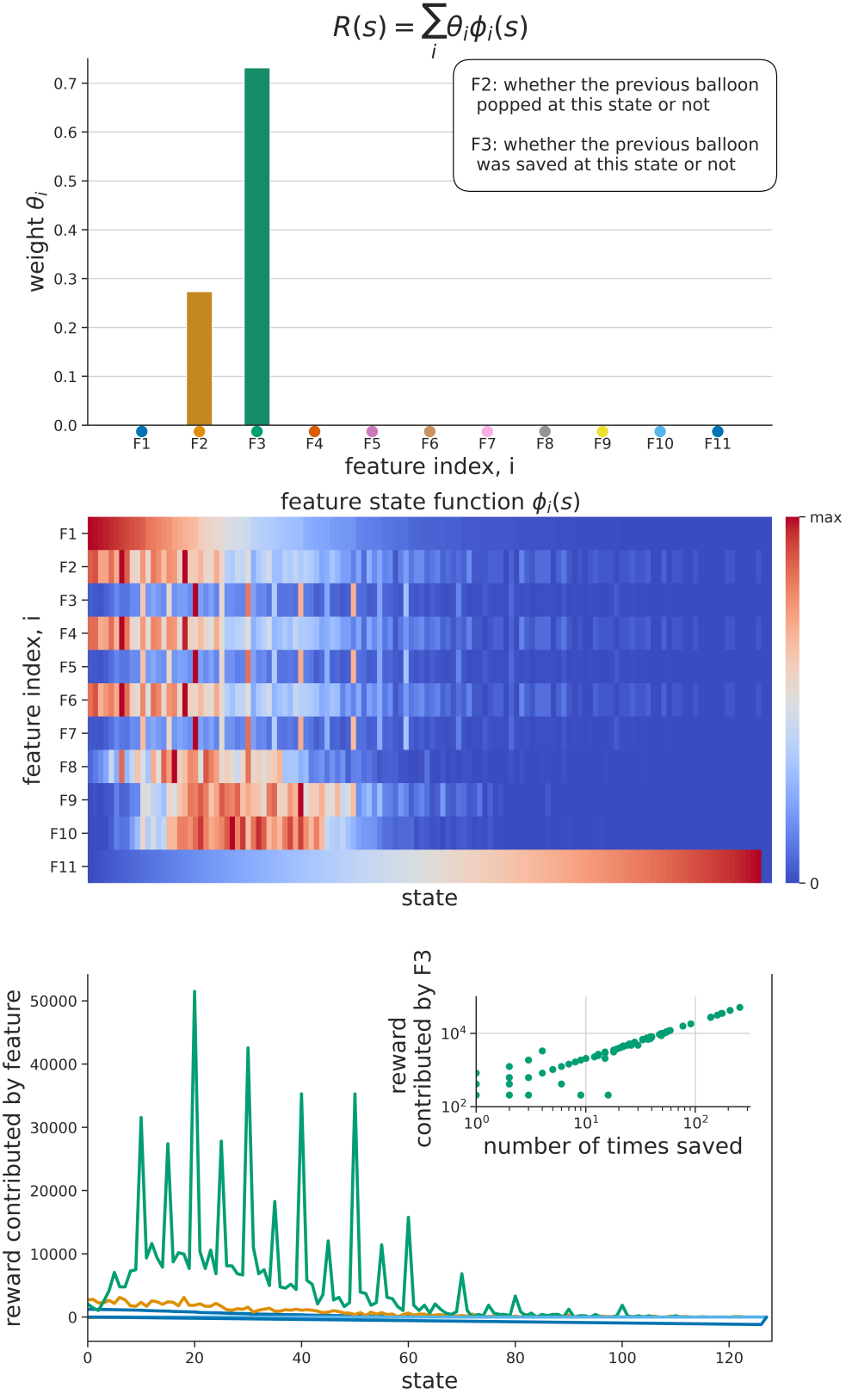

Figure 4. **Top**: Estimated reward weights recovered by the IRL algorithm from human data. **Middle**: Distribution of each feature over the state space. Red regions represent states with higher density. **Bottom**: Estimated reward distribution over states. This plot is obtained by taking the dot product between the weights and features.

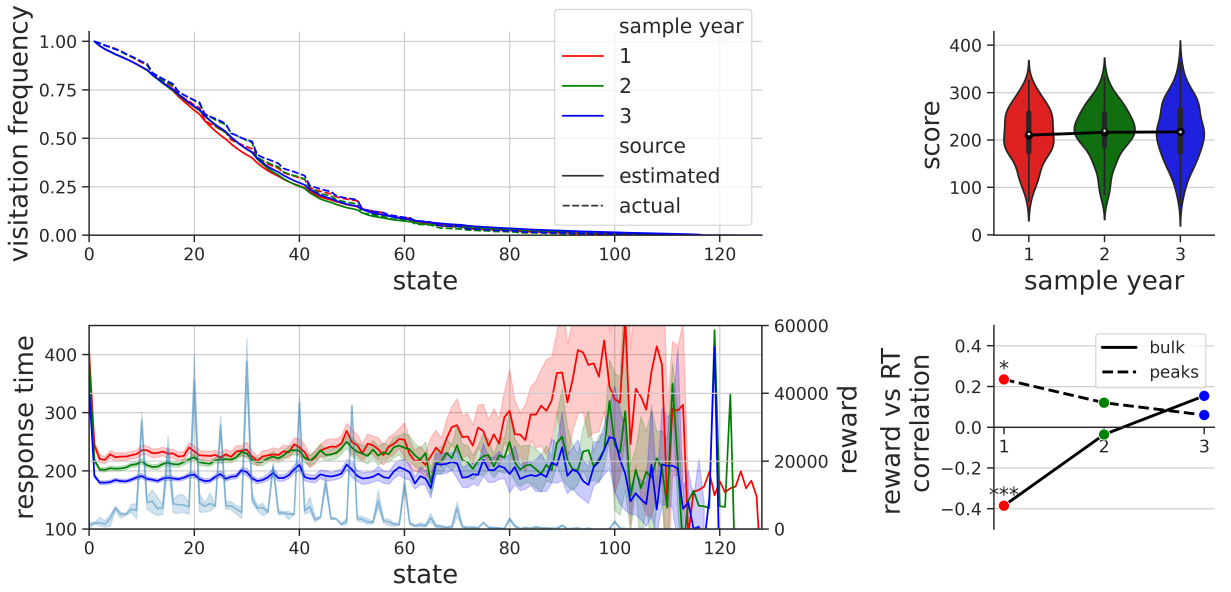

Figure 5.

heightened risk-seeking in a probabilistic gambling task. *Cognitive development*, 25(2):183–196, 2010.

[6] Bernd Figner, Rachael J Mackinlay, Friedrich Wilkening, and Elke U Weber. Affective and deliberative processes in risky choice: age differences in risk taking in the columbia card task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(3):709, 2009.

[7] Quanying Liu, Haiyan Wu, and Anqi Liu. Modeling and interpreting real-world human risk decision making with inverse reinforcement learning. *CoRR*, abs/1906.05803, 2019. URL http://arxiv.org/abs/1906.05803.

[8] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[9] Laura K Wolf, Nicholas D Wright, Emma J Kilford, Raymond J Dolan, and Sarah-Jayne Blakemore. Developmental changes in effects of risk and valence on adolescent decision-making. *Cognitive development*, 28(3):290–299, 2013.

[10] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

Supplemental Materials

Feature index	Value set	Description
1	$\{0, 1, \ldots, k-1\}$	# of previous visits to $\boldsymbol{s}$
2	$\{0, 1\}$	$\boldsymbol{s}$ burst in trial $k-1$
3	$\{0, 1\}$	$\boldsymbol{s}$ saved in trial $k-1$
4	$\{0, 1\}$	$\boldsymbol{s}$ burst in trial $k-2$
5	$\{0, 1\}$	$\boldsymbol{s}$ saved in trial $k-2$
6	$\{0, 1\}$	$\boldsymbol{s}$ burst in trial $k-3$
7	$\{0, 1\}$	$\boldsymbol{s}$ saved in trial $k-3$
8	$\{0, 1\}$	$\boldsymbol{s}$ is average burst state
9	$\{0, 1\}$	$\boldsymbol{s}$ is average save state
10	$\{0, 1\}$	$\boldsymbol{s}$ is average end state
11	$\boldsymbol{s}$	current state

Table I. Feature functions of current state, $\boldsymbol{s}$, in current trial, k.

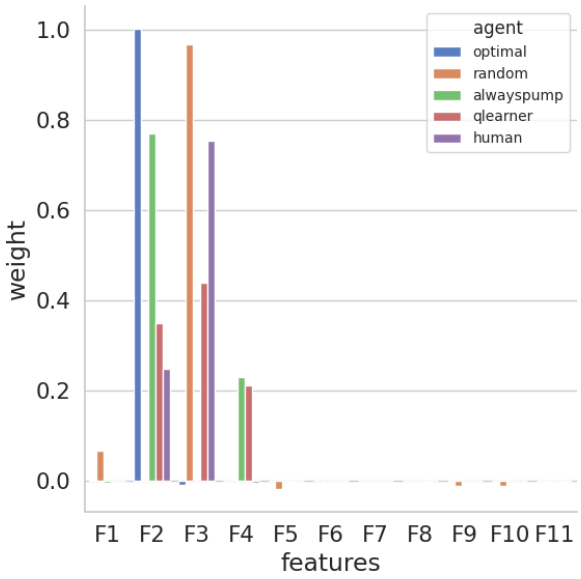

Figure S1. Learned normalized weights. The weights recovered by the MaxEnt IRL algorithm applied to 5 different agents.