Do we use the Right Measure? Challenges in Evaluating Reward Learning Algorithms

Nils Wilde
Department of Cognitive Robotics
Delft University of Technology Netherlands
n.wilde@tudelft.nl

Javier Alonso-Mora
Department of Cognitive Robotics
Delft University of Technology Netherlands
n.wilde@tudelft.nl

Abstract: Reward learning is a highly active area of research in human-robot interaction (HRI), allowing a broad range of users to specify complex robot behaviour. Experiments with simulated user input play a major role in the development and evaluation of reward learning algorithms due to the availability of a ground truth. In this paper, we review measures for evaluating reward learning algorithms used in HRI, most of which fall into two classes. In a theoretical worst case analysis and several examples, we show that both classes of measures can fail to effectively indicate how good the learned robot behaviour is. Thus, our work contributes to the characterization of sim-to-real gaps of reward learning in HRI.

Keywords: Human Robot Interaction, Reward Learning

1 Introduction

Researchers in human-robot interaction (HRI) study how robot behaviour can be adapted to the end-users' preferences. To achieve this, intelligent autonomous robotic systems often do not receive explicit instructions, but instead optimize a reward function that encapsulates how the robot should accomplish its tasks. Defining such reward functions is challenging. Therefore, interactive learning frameworks have been designed that allow a broader range of users to transfer their preferences into a reward function [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. In those frameworks, different modes of interaction have been studied, including pair-wise comparisons of trajectories [2, 3, 4, 5, 6, 7, 8, 15], rankings of trajectories [16, 17], and feedback [18, 11, 13, 19], among others.

The development and validation of efficient algorithms for reward learning requires numerical experiments where the ground truth reward function is known. Since the reward function of real users is in general unknown, this usually involves simulated users, *i.e.,* sampling potential user reward functions and then generating user input following a behaviour model. To evaluate algorithm performance various measures are used, most of which fall into two classes: *parameter-based* and *reward-based* measures. The former capture how close the learned parameters of the user's reward function are to the ground truth parameters, while the second class considers the reward that the user assigns to the learned solution.

We show that both classes have significant shortcomings: parameter-based measures do not give a direct indication about the reward collected, *i.e.,* how *good* the learned behaviour is in specific instances, while reward based measures do not necessarily translate well between related problem instances, *e.g.,* between training and test environments. These observations shed light on limitations of evaluation methods for reward learning algorithms in HRI: i) Even similar (but not identical) parameters do not guarantee that the learned robot behaviour is close to optimal. ii) Reward-based measures might indicate strong performance in training environments, yet the performance in test environments can be much poorer, even when the environments are relatively similar. Thus, both classes of measures do not effectively indicate how good the learned robot behaviour is in general.

Related Work: Parameter- and reward-based measures are used extensively in reward learning in HRI [2, 4, 5, 20, 21, 22, 23, 24, 25, 26]. Yet, their respective shortcomings are not widely discussed.

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

To address the transfer problem between training and testing, it is common in machine learning to use several testing instances [27]. This practise is also applied in reinforcement learning (RL) and inverse reinforcement leraning (IRL) [28, 29]. The authors of Fu et al. [29] use an adversarial RL framework to obtain a reward function that yields robust behaviour in test instances. The relation of inaccurate parameters and optimality of the corresponding solution leads to the observation that even inaccurate reward functions can yield an optimal RL-policy [29, 30, 31]. Similarly, Wilde et al. [9] discretize the parameter space of a graph-based planner into subsets called *equivalence regions* such that all parameters from the same subset yield the same path. Brown et al. [32] recently proposed *value alignment* for Markov Decision Processes (MDP) to capture if the robot behaviour corresponds to a user's preference, avoiding the pitfalls of parameter-based measures. Gleave et al. [31] propose a measure for evaluating reward functions in RL without requiring the costly computation of a policy. In contrast to these works, our paper does not focus on a specific problem domain such as RL or graph search. Instead, we study shortcomings of learning measures used in HRI, describe their theoretical worst-case and show different characteristics in numerical experiments.

Contributions: In this paper, we provide a review of measures used in HRI for how well a robot has learned a user's reward function, followed by an extensive analysis of their limitations. In particular, we study exemplary cases highlighting the shortcomings of the two main reward learning measures. We then provide a theoretical analysis, showing that both measures can be arbitrarily bad indicators for the actual performance. In numerical experiments for three different robot planning problems, we highlight that the issue is prevalent in various domains.

1.1 Reward Learning Formalism

Let I be the instance of a robot planning problem, defined by a robot's state and action space, an initial state, a set of goal states, and a collection of constraints. In HRI, reward functions are often used to describe how well a robot trajectory fits a user's preferences. Thus, we consider a reward function R taking a robot trajectory $\mathcal{T}$ as an input and assigning a real value. Usually, R is assumed to be a weighted sum of features [33, 23, 2]. Let $\mathcal{T}$ be a robot trajectory, and $\boldsymbol{\phi}(\mathcal{T}) = [\phi_1(\mathcal{T}) \ \dots \ \phi_n(\mathcal{T})]$ be some predefined feature functions. Then the reward of $\mathcal{T}$ is

$$R(\mathcal{T}, \boldsymbol{w}^{\texttt{user}}) = \boldsymbol{w}^{\texttt{user}} \cdot \boldsymbol{\phi}(\mathcal{T}), \tag{1}$$

where $\boldsymbol{w}^{\texttt{user}}$ is a weight vector, expressing how a particular user values these features. Without loss of generality, we assume the weights to be positive. The objective is that a robot executes trajectories of maximum reward, *i.e.,* computes

$$\mathcal{T}^{\texttt{user}} = \arg\max_{\mathcal{T}} R(\mathcal{T}, \boldsymbol{w}^{\texttt{user}}) \tag{2}$$

for any instance I of the planning problem. However, in many applications, it is difficult to obtain the weights $\boldsymbol{w}^{\texttt{user}}$ from a particular user. Reward learning is the problem of interactively learning these weights in order to solve (2). This can be informally stated as follows:

Problem 1 (Reward Learning)**.** Given is a user with hidden preference $\boldsymbol{w}^{\texttt{user}}$, some mode of user interaction $\mathcal{M}$, and K iterations for querying the user. Using the interaction, the robot needs to learn the best possible estimate $\boldsymbol{w}'$ of the user's weight vector $\boldsymbol{w}^{\texttt{user}}$.

2 Review of Reward Learning Measures

Problem 1 is ambiguous about what defines a *good* estimate $\boldsymbol{w}'$. In simulation experiments ground truth user weights $\boldsymbol{w}^{\texttt{user}}$ are available, allowing for a direct comparison to optimal. The HRI community uses different measures, which can be divided into two primary categories: *reward-based* and *parameter-based*. We will briefly review the most common measures from the literature, followed by a discussion about the fundamental difference of reward-based and parameter-based measures.

2.1 Reward-Based Measures:

Reward-based measures directly evaluate estimated weights $\boldsymbol{w}'$ based on how well the corresponding optimal trajectory $\mathcal{T}'$ solves the principal problem in (2).

Returned Reward: The most direct reward-based measure is to consider the reward $R(\mathcal{T}', \boldsymbol{w}^{\texttt{user}})$ some trajectory $\mathcal{T}'$ collects for user weights $\boldsymbol{w}^{\texttt{user}}$. This measure has been used in [20, 21, 22], among many others.

Relative Reward: To measure closeness to optimal, the *Relative Reward* [15, 26] takes the ratio of the reward $\boldsymbol{w}^{\texttt{user}}$ assigns to the learned solution $\mathcal{T}'$ over the reward assigned to $\mathcal{T}^{\texttt{user}}$, *i.e.,*

$$R^{\texttt{rel}}(\boldsymbol{w}', \boldsymbol{w}^{\texttt{user}}) = \frac{R(\mathcal{T}', \boldsymbol{w}^{\texttt{user}})}{R(\mathcal{T}^{\texttt{user}}, \boldsymbol{w}^{\texttt{user}})}. \tag{3}$$

In case the reward function takes negative values, the ratio is inverted. When rewards can be positive and negative, a normalization is needed.

Regret: In some cases, it might be favourable to use a negative measure, *i.e.,* where 0 corresponds to the best achievable value. This is capture by the *Relative Regret* [5], which is $\texttt{Reg}^{\texttt{rel}} = 1 - R^{\texttt{rel}}(\boldsymbol{w}', \boldsymbol{w}^{\texttt{user}})$. Alternative to the ratio, the *Absolute Regret* is formulated as a difference [26, 23, 25], *i.e.,* $\texttt{Reg}^{\text{abs}} = R(\mathcal{T}^{\texttt{user}}, \boldsymbol{w}^{\texttt{user}}) - R(\mathcal{T}', \boldsymbol{w}^{\texttt{user}})$. This is a convex function of $\boldsymbol{w}^{\texttt{user}}$ [26].

2.2 Parameter-Based Measures:

This group of measures also requires ground truth weights $\boldsymbol{w}^{\texttt{user}}$, and measures the similarity of the estimated weights $\boldsymbol{w}'$ and the optimal weights $\boldsymbol{w}^{\texttt{user}}$.

Alignment: The alignment measure [2, 4, 34] captures how similar the learned weights are to the user weights by measuring the angle between the vectors:

$$\alpha(\boldsymbol{w}', \boldsymbol{w}^{\texttt{user}}) = \frac{\boldsymbol{w}' \cdot \boldsymbol{w}^{\texttt{user}}}{||\boldsymbol{w}'||\,||\boldsymbol{w}^{\texttt{user}}||}. \tag{4}$$

Mean squared weight error: Similar to alignment, other works use the mean squared error (MSE) of estimated weights and user weights [23, 24].

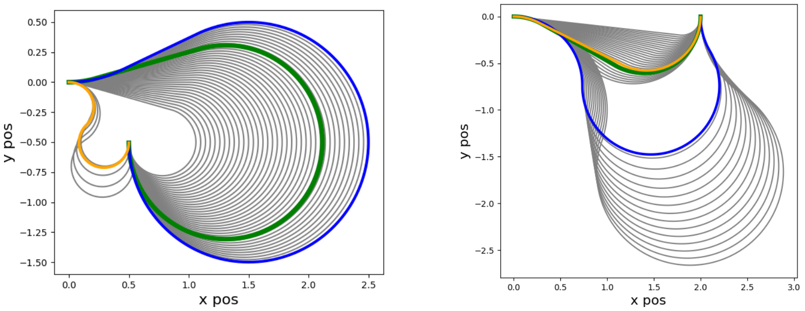
2.3 Other Measures

For completeness we also briefly review a third category used in learning from choice frameworks, which we label as *predictive measures*. There are two measures in this category, **log-likelihood** [16, 3, 15], and prediction **accuracy** [3]. Both measures describe how well a learned probability distribution over weights $\boldsymbol{w}$ allows for predicting how a user would choose between two trajectories. These measures do not require the ground truth weights, but only a user's response to a validation set consisting of choice queries, which makes them suitable for evaluating user studies. However, prediction measures are indirect and depend on the validation set.

2.4 Comparison of Measures

Parameter-based and reward-based measures take different perspectives on describing how well some estimate $\boldsymbol{w}'$ describes a user reward function with weights $\boldsymbol{w}^{\texttt{user}}$. *Reward-based* measures consider the robot trajectory that a planner or policy returns when optimizing for weights $\boldsymbol{w}'$, and compute the reward assigned to it by the user $\boldsymbol{w}^{\texttt{user}}$. This directly captures how well the robot is able to solve problem (2). However, this statement only relates to *one specific problem instance*. For instance, a robot might learn a reward function for how to carry dishes around the kitchen, where the features are trajectory length and risk of collision. When trained for a specific planning problem in a specific kitchen, the reward-based measures then describe how well the robot is able to solve that task, but give no direct information about how well the robot would operate in a different kitchen.

In contrast, *parameter-based* measures describe how well the reward *function* is estimated. These measures are universal, *i.e.,* independent of the instance. When capturing $\boldsymbol{w}^{\texttt{user}}$ accurately, a robot would be able to compute $\mathcal{T}^{\texttt{user}}$ in any problem instance with the same features. Yet, parameter-based measures also have a major drawback: they do not consider the robot's planner, *i.e.,* the mapping from weights to optimal trajectory. Thus, there might be no relation to the collected reward - and thus no information about how well the robot will solve (2), unless $\boldsymbol{w}' = \boldsymbol{w}^{\texttt{user}}$.



(a) Dubins instance 1.

(b) Dubins instance 2.

Figure 1: Different Dubins paths with three features. Green shows a user's optimal trajectory $\mathcal{T}^{\text{user}}$, blue and orange two different estimates $\mathcal{T}^{A}$ and $\mathcal{T}^{B}$. Grey shows other Dubins paths for different radii.

(a) Instance 1, Fig. 1a.

	R	R^{rel}	Reg^{abs}	α	MSE
$\boldsymbol{w}^{\text{user}}$	-5.84	1	0	1	.01
$\boldsymbol{w}^{A}$	-5.92	.98	.08	.98	.01
$\boldsymbol{w}^{B}$	-6.36	.92	.52	.98	.01

(b) Instance 2, Fig. 1b.

	R	R^{rel}	Reg^{abs}	α	MSE
$\boldsymbol{w}^{\text{user}}$	-3.39	1	0	1	.01
$\boldsymbol{w}^{A}$	-3.56	.95	.17	.98	.01
$\boldsymbol{w}^{B}$	-3.39	.1	0	.98	.01

Table 1: Estimation measures for the Dubins example.

3 Exemplary Fail-Cases

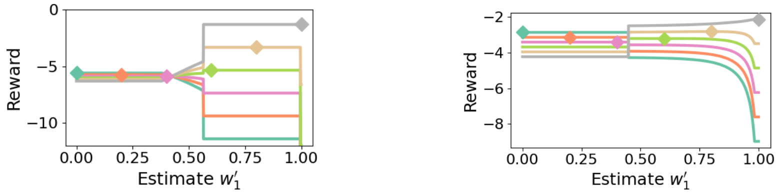
Using a simple motion planning task we now illustrate the weaknesses of parameter and reward based measures. A robot is navigating in 2D space with constant velocity following Dubins paths, where the radius can be chosen within some bounds. There are two features: path length, and integral square jerk (IS jerk) – the first captures the time-efficiency while the second measures discomfort or risk (*Note: to obtain a maximization problem, we take the negative of these values*).

Single-Scenario Example: We arbitrarily choose a user with preference $\boldsymbol{w}^{\text{user}} = [.5, .5]$, and two estimates $\boldsymbol{w}^{A} = [.4, .6]$, and $\boldsymbol{w}^{B} = [.6, .4]$. We illustrate the corresponding optimal trajectories in Figure 1a and summarize the resulting measures in Table 1(a). The parameter-based measures evaluate $\boldsymbol{w}^{A}$ and $\boldsymbol{w}^{B}$ as equally good estimates of $\boldsymbol{w}^{\text{user}}$. However, $\boldsymbol{w}^{A}$ collects more reward than $\boldsymbol{w}^{B}$, making it the better solution. This observation demonstrates that the alignment metric is unaware of the sensitivity of the optimization to find the optimal trajectory given some $\boldsymbol{w}$. Thus, alignment can fail to capture how well estimated weights can allow the robot to compute trajectories that fit a user's preference.

Multi-Scenario Example: We now study the case where a robot has to perform multiple similar tasks. Hence, we consider a second planning instance in Figure 1b with a different goal location. This serves as a test case, while we use the same $\boldsymbol{w}^{\text{user}}$ and estimates $\boldsymbol{w}^{A}$ and $\boldsymbol{w}^{B}$ as before. In practise, a robot might have obtained these estimates from reward learning in the first instance. A good estimate of $\boldsymbol{w}^{\text{user}}$ should then also achieve high reward in the test case. The measure values are shown in Table 1(b). Naturally, the alignment and MSE yield the same value as earlier since they only depend on the weights, not on the planning instance or trajectory. However, the reward measures show that, while $\boldsymbol{w}^{B}$ was a poorer estimate than $\boldsymbol{w}^{A}$ in the first example, the places are now reversed. Hence, good values of a *reward-based* metric do not necessarily translate between instances, *e.g.,* from training to test cases.

4 Theoretical Analysis

In addition to the previous motion planning tasks with a Dubins model, we present more rigorous theoretical results. For brevity, the proofs can be found in the supplementary material.



(a) Rewards for Dubins instance 1.

(b) Rewards for Dubins instance 2.

Figure 2: Example reward functions $r_{\boldsymbol{w}^{\text{user}}}(\boldsymbol{w}')$ for the Dubins problem from Figure 1. In each plot, we pick 5 different $\boldsymbol{w}^{\text{user}}$, uniformly spaced, and plot the respective rewards $r_{\boldsymbol{w}^{\text{user}}}(\boldsymbol{w}')$. The markers on each line show the location of the optimal estimate, *i.e.,* where $\boldsymbol{w}' = \boldsymbol{w}^{\text{user}}$.

4.1 Worst-Case Study

First, we show that for any non-optimal alignment or MSE error the collected reward under the true weights $\boldsymbol{w}^{\text{user}}$ can be arbitrarily far from optimum.

Theorem 1 (Unbounded Reward Difference)**.** Let $\boldsymbol{w}^{\text{user}}$ be a user weight, and $\boldsymbol{w}'$ be an estimate, where the alignment is $\delta \leq \alpha(\boldsymbol{w}', \boldsymbol{w}^{\text{user}}) < 1$ for some $\delta < 1$. The difference in reward $R(\mathcal{T}^{\text{user}}, \boldsymbol{w}^{\text{user}}) - R(\mathcal{T}', \boldsymbol{w}^{\text{user}})$ is unbounded.

Theorem 1 shows that even when the alignment is arbitrarily close to 1, it does not allow for claims on how much reward is collected, compared to optimal. Following the same proof the result extends to the second parameter-based measure, the MSE. Next, we study the multi-scenario case to show that the reward-based measures do not translate from test to training scenarios. Let I^{Train} be a training and I^{Test} a test instance. For weights $\boldsymbol{w}'$, $\boldsymbol{w}^{\text{user}}$, we use $R^{\texttt{rel}}_{\texttt{train}}(\boldsymbol{w}', \boldsymbol{w}^{\text{user}})$ to denote the relative reward collected in I^{Train} and similarly $R^{\texttt{rel}}_{\texttt{test}}(\boldsymbol{w}', \boldsymbol{w}^{\text{user}})$ for the test instance I^{Test}.

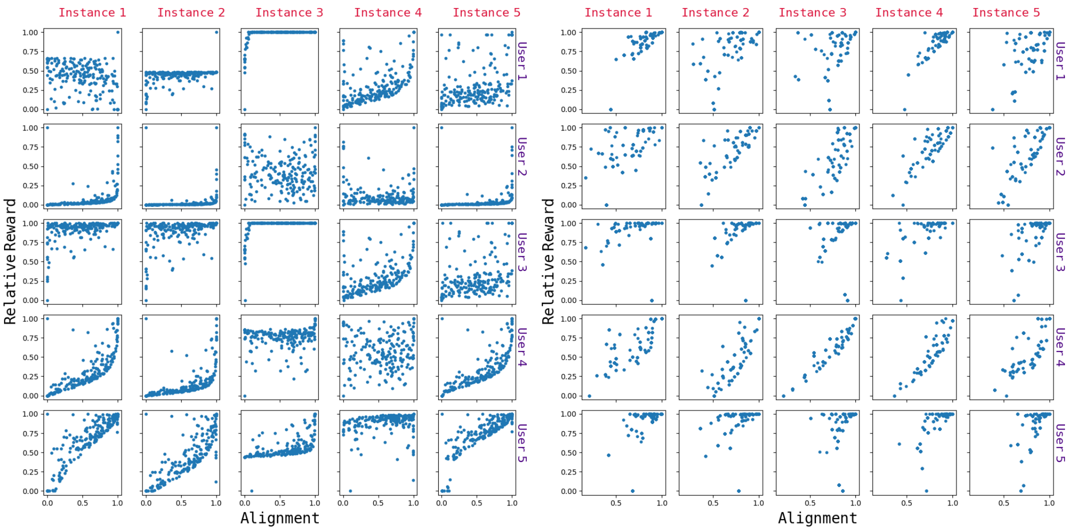
Theorem 2 (Unbounded Test Error)**.** Let $\boldsymbol{w}^{\text{user}}$ be a user weight, and $\boldsymbol{w}'$ be an estimate. Further, let I^{Train} be a training instance, where the relative reward $R^{\texttt{rel}}_{\texttt{train}}(\boldsymbol{w}', \boldsymbol{w}^{\text{user}})$ is taking values in $[\delta, 1]$ for some $\delta < 1$. There exist test instances I^{Test} where the relative reward $R^{\texttt{rel}}_{\texttt{test}}(\boldsymbol{w}', \boldsymbol{w}^{\text{user}})$ has no tighter lower bound than 0.

The two theorems show that in a worst case the observed shortcomings can cause both classes of measures to be arbitrarily poor indicators of the actual performance.

4.2 Non-linear Characteristics of Reward Functions

Given the exemplary cases and the worst-case analysis, we briefly characterize the underlying relationship of weights and rewards. Using the convention that $\mathcal{T}'$ is the optimal trajectory for weights $\boldsymbol{w}'$ we can write the collected reward when optimizing for estimate $\boldsymbol{w}'$ as $r(\boldsymbol{w}^{\text{user}}, \boldsymbol{w}') = \boldsymbol{w}^{\text{user}} \cdot \boldsymbol{\phi}(\mathcal{T}')$. This is a linear function of $\boldsymbol{w}^{\text{user}}$ for any fixed $\boldsymbol{w}'$. Yet, maybe surprisingly, for any fixed $\boldsymbol{w}^{\text{user}}$ it is in general a non-linear function of the estimate $\boldsymbol{w}'$. That is, the features $\boldsymbol{\phi}(\mathcal{T}')$ are often not linearly dependent on the weights $\boldsymbol{w}'$. This is related to the sensitivity of optimization problems – small changes in $\boldsymbol{w}'$ can lead to the same solution. Indeed, if trajectories are computed with a discrete planner, then almost all trajectories outputted by the planner are an optimal solution to not just a single weight, but a set of weights. Thus, the function $r(\boldsymbol{w}^{\text{user}}, \boldsymbol{w}')$ is piece-wise constant over $\boldsymbol{w}'$ [9]. Yet, this phenomenon also applies in continuous space. For instance, when planning around an obstacle, multiple weights can lead to the tightest feasible trajectory that takes the detour *around* the obstacle; only for a high enough weight on minimizing trajectory length, the planner will switch to the other side of the obstacle. Different weights leading to the same optimal solution is also related to *reward-ambiguity* in RL [30, 29].

We illustrate the non-linearity of $r(\boldsymbol{w}^{\text{user}}, \boldsymbol{w}')$ in Figure 2 using the planning examples from Section 3, where we fix several values for $\boldsymbol{w}^{\text{user}}$. Since the problem is only two-dimensional, we can set $w_2 = 1 - w_1$, allowing us to plot $r(\boldsymbol{w}^{\text{user}}, \boldsymbol{w}')$ over the scalar values of w'_1. We make two key observations: i) $r(\boldsymbol{w}^{\text{user}}, \boldsymbol{w}')$ is non-linear in $\boldsymbol{w}'$ and and can exhibit jumps, which causes the shortcomings of parameter-based measures. ii) $r(\boldsymbol{w}^{\text{user}}, \boldsymbol{w}')$ shares very little similarities between the two instances, resulting in the transfer problem between training and testing.



(a) Dubins experiment.

(b) Driver experiment.

Figure 3: Examples for the relationship between *alignment* and *relative reward*. In each plot, the relative reward values have been normalized.

	Dubins				Driver				Server			
	Mn	Md	SD	Min/Max	Mn	Md	SD	Min/Max	Mn	Md	SD	Min/Max
Pears.	.41	.43	.28	-.3/ .96	.45	.50	.32	-.39/ .96	.45	.46	.18	-.06/ .87
Spear.	.5	.56	.29	-.36/ .97	.56	.59	.23	-.17/ .97	.42	.42	.18	-.06/ .85

Table 3: Correlations between alignment and relative reward. Correlation values: Pears. – Pearson correlation coefficient, Spear.– Spearman Rank coefficient. Statistics: Mn–mean, Md – median, SD standard deviation.

5 Numerical Results

To validate that the described fallacies of the measures do not only arise in constructed examples, we conduct extensive numerical experiments with randomly generated instances and user weights.

Robot planning problems: We consider three different robot experiments: i) A mobile navigation problem using a Dubin's car similar to Figure 1 but with three features: trajectory length, jerk and the closeness to humans present in the environment. ii) The driver simulation previously used in [2, 4, 5, 15, 32], among others. This problem has four features: keeping speed, closeness to another vehicle, staying in a lane, and orientation on the road. iii) A manipulator robot serving drinks. The experiment has eight features, describing the choice of drink as well as how the robot moves over a plate and stove. Trajectories were generated on a real-world system by the authors of [15].

Generation of user weights and estimates For each experiment, we generate a set Ω of 100 uniformly random weights vectors in $[0,1]^n$ (where n is the number of features). User weights $\boldsymbol{w}^{\texttt{user}}$ are picked uniformly random from Ω. We do not conduct any reward learning. Instead, all weights $\boldsymbol{w}' \in \Omega$ are considered as estimates of $\boldsymbol{w}^{\texttt{user}}$, which could have been the result of learning.

Dependent Measures: For each pair $(\boldsymbol{w}^{\texttt{user}}, \boldsymbol{w}')$ we compute different learning measures. We characterize the relationship between measures using the Pearson correlation and Spearman rank coefficients. The Pearson coefficient describes a linear correlation – a strong linear relationship would be the best possible result. The Spearman coefficient measures monotonicity.

	Dubins				Driver				Server			
	Mn	Md	SD	Min, Max	Mn	Md	SD	Min, Max	Mn	Md	SD	Min, Max
Pears.	.44	.47	.23	-.5/.97	.48	.47	.25	-.57/.90	.80	.91	.26	-.21/1.00
Spear.	.45	.43	.26	-.55/.97	.76	.83	.19	-.07/.95	.80	.93	.28	-.17/1.00

Table 4: Correlations between relative reward in training and testing.

5.1 Analysis of Alignment-Reward Relation

First, we investigate how alignment relates to solving the principal problem in equation (2). For any pair $(\boldsymbol{w}^{\texttt{user}}, \boldsymbol{w}')$ we compute alignment and relative reward[1]. Figure 3 shows exemplary results for 5 users, across 5 different problem instances. Overall, the relationship between alignment and relative reward varies drastically between instances as well as between users. The results fall into five categories:

1. *Approximately Linear:* In some instances alignment is a good indication of the collected reward. Examples are Dubins: User 5, Instances 1 and 5, as well as Driver: Users 2 and 4, Instances 2-5.
2. *Underestimates / Simple Instances:* In some cases, most samples with alignments $> .5$ collect close to optimal reward. Thus, the different sampled solutions are similar to optimal solutions for these users, resulting in easy problem instances. Here, the alignment is a drastic underestimate of the actually collected reward. Examples are Dubins: Users 1 and 3,Instance 3, Driver: User 3 and 5, all instances expect 3.
3. *Overestimates:* Some plots show a monotone trend that becomes steeper for higher alignments. In most such cases the relative reward is small even for high alignment values. Thus, the collected reward is overestimated. In case of Dubins, User 2, Instance 1 or 2, the alignment values of $\approx .95$ correspond to less than $.25$ relative reward, *i.e.,* highly suboptimal solutions. Other examples are Dubins: User 4 Instances 1, 2 and 5.
4. *Unstructured:* In various instances, alignment and reward have no, or at most a weak correlation, *i.e.,* alignment gives almost no reliable information on the collected reward.

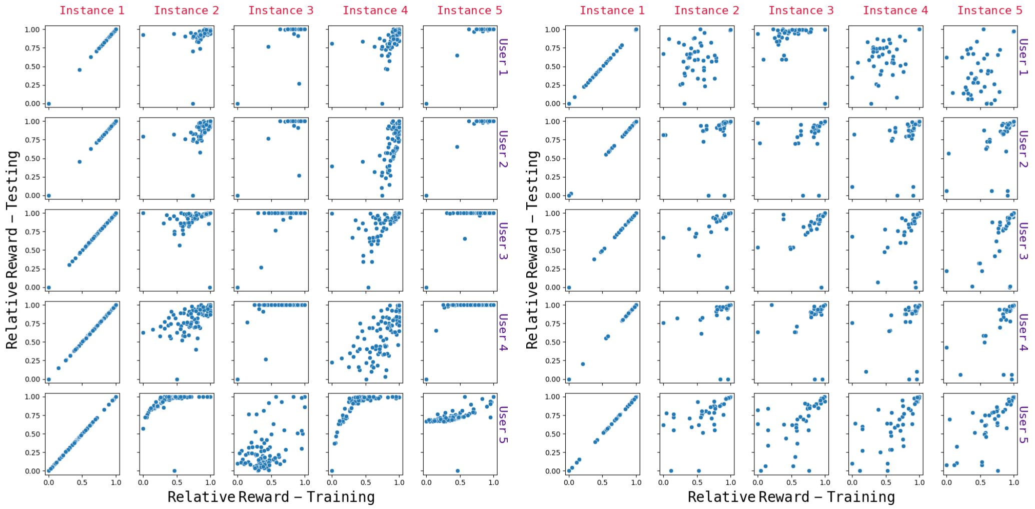
We repeated the experiments for 40 random seeds for Ω, yielding 1000 user-instance pairs, numerical results are shown in Table 3. For all three planning problems the Pearson correlations are weak and have high standard deviations. This strongly supports the observations from the plots: Across instances and users there is no consistent linear trend between alignment and reward. The Spearman coefficients are slightly higher for Dubins and Driver, yet not high enough to indicate a strong monotonic relationship. In summary, the experiments show that alignment is not a reliable indicator on how much reward is collected, *i.e.,* how well some estimate $\boldsymbol{w}'$ solves the principal problem (2).

5.2 Analysis of Reward between Training and Testing

Next, we study the relation of relative reward between training and test instances. We fix one instance as a *training* instance $I^{\texttt{Train}}$, and compute the relative reward with respect to $\boldsymbol{w}^{\texttt{user}}$ for all $\boldsymbol{w}' \in \Omega$. For the same samples, we compute the relative reward in a set of *test* instances. Figure 4 shows exemplary results: Each row corresponds to a fixed random user, and each column is a different instance with the first column being the training, and the others the test instances.

We observe a large variability in the relationship between the relative rewards collected in training and testing, with similar archetypes of correlations as in the first experiment. For Driver we notice that in Instances 3 and 5 almost all samples are close to optimal for user 1-4, showing that these instances are trivial for some, but not all users. Again, we repeated the experiment for 40 random seeds and report the statistic in Table 4. For the Dubins and Driver experiment, we observe only weak correlations – the reward collected in training is not a reliable indicator for the reward collected in other instances. However, the Server task exhibits strong Pearson and Spearman correlations. Indeed, for

[1] The relative value is favourable over returned reward or absolute regret since it takes values in the unit interval. Further, we use relative reward instead of relative regret such that the optimum is at 1, akin to alignment.



(a) Dubins experiment.

(b) Driver experiment.

Figure 4: Examples for the relationship between relative reward in *Training* and *Testing*. In each plot, the relative reward values have been normalized.

many instances, the relationship follows a strong linear trend allowing for better predictions (example plots can be found in the supplementary material). Nonetheless, the high standard deviations and very small min values reveal that this does not extend to all instances and users.

In summary, we observe that the relative reward does not necessarily translate from training to test instances. Thus, high rewards in training do not necessarily yield good performance in test instances.

6 Discussion

Summary: In this paper, we reviewed measures for evaluating reward learning algorithms when ground truth weights $\boldsymbol{w}^{\texttt{user}}$ are available, and characterized the two main classes of measures: parameter-based and reward-based. Through i) illustrative examples, ii) theoretical worst-case analysis, and iii) experiments with various robot planning problems, we have shown that both classes are not reliable indicators for how well the robot has adapted to a user's preferences.

Limitations: While our analysis describes theoretical worst-cases for both classes of measures, it does not derive characteristics of feature functions under which the shortcomings are more severe, or where a tighter worst-case bound could be derived. Further, our work is mostly concerned with identifying and describing these issues, but provide no detailed study of how prevalent the they are in published work. Moreover, the paper relies on relatively simple planning problems for theoretical studies, leaving it ambiguous how severe the issues are in more complex or real-world settings.

Future Work: Our analysis raises the question about a better way to evaluate reward learning algorithms. The transfer problem can be addressed by using several test instances for reward-based measures, which is common in machine learning. This approach could be adapted to actively or adversarially selecting test instances similar to [29]. Unlike random test instances this could yield a bound on the error in *any* test instance, *i.e.,* soften the negative result of Theorem 2. In Section 4.2 we showed that the relation from weights to features and thus rewards for a fixed $\boldsymbol{w}^{\texttt{user}}$ is non-linear. Yet, this depends on the choice of features. Future work could investigate if features can be modified to mitigate the two shortcomings described in this paper. However, it is unclear if there exists a principal way, or if this would require creating proxy-features for individual instance. Finally, post-processing estimates could make them more robust towards the observed fallacies. Even when collecting high reward in training, the estimated parameters are likely not accurate. This allows us to modify the estimate by solving a bi-objecitve optimization problem: the first objective is that the new estimate collects as much reward as the initial estimate in the training. The second objective then solves a max-min robust optimization problem for the test instance.

Acknowledgments

Acknowledgements: This research is partially supported by the European Union's Horizon 2020 research and innovation program under Grant 101017008.

References

[1] H. J. Jeon, S. Milli, and A. D. Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. In *Advances in Neural Information Processing Systems (NIPS)*, Dec. 2020.

[2] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2017.

[3] E. Biyik, N. Huynh, M. J. Kochenderfer, and D. Sadigh. Active preference-based gaussian process regression for reward learning. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2020.

[4] E. Biyik, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh. Asking easy questions: A user-friendly approach to active reward learning. In *Proceedings of the 3rd Conference on Robot Learning (CoRL)*, 2019.

[5] N. Wilde, D. Kulić, and S. L. Smith. Active preference learning using maximum regret. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10952–10959, 2020.

[6] C. Basu, M. Singhal, and A. D. Dragan. Learning from richer human guidance: Augmenting comparison-based learning with feature queries. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 132–140, 2018.

[7] R. Holladay, S. Javdani, A. Dragan, and S. Srinivasa. Active comparison based learning incorporating user uncertainty and noise. In *RSS Workshop on Model Learning for Human-Robot Communication*, 2016.

[8] K. Li, M. Tucker, E. Biyik, E. Novoseller, J. W. Burdick, Y. Sui, D. Sadigh, Y. Yue, and A. D. Ames. Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes. In *International Conference on Robotics and Automation (ICRA)*, May 2021.

[9] N. Wilde, D. Kulić, and S. L. Smith. Bayesian active learning for collaborative task specification using equivalence regions. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1691–1698, Apr. 2019. ISSN 2377-3766.

[10] N. Wilde, A. Blidaru, S. L. Smith, and D. Kulić. Improving user specifications for robot behavior through active preference learning: Framework and evaluation. *The International Journal of Robotics Research (IJRR)*, 39(6):651–667, 2020.

[11] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan. Learning robot objectives from physical human interaction. In *Proceedings of the 1st Conference on Robot Learning (CoRL)*, pages 217–226. PMLR, 2017.

[12] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. Learning reward functions by integrating human demonstrations and preferences. In *Proceedings of Robotics: Science and Systems (RSS)*, June 2019.

[13] M. Li, A. Canberk, D. P. Losey, and D. Sadigh. Learning human objectives from sequences of physical corrections. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[14] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler. Human preferences for robot-human hand-over configurations. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1986–1993. IEEE, 2011.

[15] N. Wilde, E. Biyik, D. Sadigh, and S. L. Smith. Learning reward functions from scale feedback. In *Proceedings of the 5th Annual Conference on Robot Learning*, pages 353–362. PMLR, 2021.

[16] V. Myers, E. Biyik, N. Anari, and D. Sadigh. Learning multimodal rewards from rankings. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, Nov. 2021.

[17] D. Brown, W. Goo, P. Nagarajan, and S. Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pages 783–792. PMLR, 2019.

[18] M. Kollmitz, T. Koller, J. Boedecker, and W. Burgard. Learning human-aware robot navigation from physical interaction via inverse reinforcement learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11025–11031. IEEE, 2020.

[19] Y. Cui and S. Niekum. Active reward learning from critiques. In *International Conference on Robotics and Automation (ICRA)*, pages 6907–6914. IEEE, 2018.

[20] D. S. Brown, W. Goo, and S. Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Proceedings of the 4th Conference on Robot Learning (CoRL)*, pages 330–359. PMLR, 2020.

[21] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4299–4307, 2017.

[22] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[23] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. *Advances in Neural Information Processing Systems (NIPS)*, 29, 2016.

[24] V. Myers, E. Biyik, N. Anari, and D. Sadigh. Learning multimodal rewards from rankings. In *Proceedings of the 5th Annual Conference on Robot Learning*, pages 342–352. PMLR, 2021.

[25] D. S. Brown, Y. Cui, and S. Niekum. Risk-aware active inverse reinforcement learning. In *Proceedings of the 2nd Conference on Robot Learning (CoRL)*, pages 362–372. PMLR, 2018.

[26] N. Wilde, A. Sadeghi, and S. L. Smith. Learning submodular objectives for team environmental monitoring. *IEEE Robotics and Automation Letters*, 7(2):960–967, 2021.

[27] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[28] S. Levine, Z. Popovic, and V. Koltun. Nonlinear inverse reinforcement learning with gaussian processes. *Advances in Neural Information Processing Systems (NIPS)*, 24, 2011.

[29] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

[30] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, volume 99, pages 278–287, 1999.

[31] A. Gleave, M. Dennis, S. Legg, S. Russell, and J. Leike. Quantifying differences in reward functions. In *International Conference on Learning Representations (ICLR)*, 2021.

[32] D. S. Brown, J. Schneider, A. Dragan, and S. Niekum. Value alignment verification. In *International Conference on Machine Learning*, pages 1105–1115. PMLR, 2021.

[33] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[34] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research (IJRR)*, 41(1):45–67, 2022.