Towards learning reward functions from user interactions

Li, Z.; Kiseleva, J.; de Rijke, M.; Grotov, A.

DOI
10.1145/3121050.3121098

Publication date
2017

Document Version
Final published version

Published in
ICTIR'17: proceedings of the 2017 ACM SIGIR International Conference on the Theory of Information Retrieval

Citation for published version (APA):
Towards Learning Reward Functions from User Interactions

Ziming Li
University of Amsterdam
Amsterdam, The Netherlands
z.li@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

Julia Kiseleva
UserSat.com & University of Amsterdam
Amsterdam, The Netherlands
j.kiseleva@uva.nl

Artem Grotov
University of Amsterdam
Amsterdam, The Netherlands
a.grotov@uva.nl

1 INTRODUCTION

Understanding and modeling user behavior is a fundamental problem for any interactive system as insight into user behavior will lead towards “proper” evaluation: what satisfies user needs and what frustrates users. We know that users have different preferences and can display different behavior [20, 32]. Despite this key lesson, the evaluation metrics in use today do not take differences in user behavior into account. Existing methods are directed towards generalization [12] rather than personalization [35]. This stops us from deciphering fine-grained user feedback. The idea of designing an interactive system that dynamically reacts to user actions by employing the reinforcement learning (RL) paradigm as proposed in Figure 1 is appealing. A key problem preventing this is that we do not know the users’ true reward functions.

Let’s consider an example of interactive system—a web search engine. Assume a user issues the query “panda” and the search engine returns a diverse search engine result page (SERP) that contains answers from various verticals: text, images, videos. If our user is a child, he will most likely click on an image result, the screen size is small or a system is expected to give the best assistants on mobile devices [17, 18] and direct answers [34], where the popularity of mobile phones, we are moving to an era of personal assistants on mobile devices [17, 18] and direct answers [34], where the screen size is small or a system is expected to give the best

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM ISBN 978-1-4503-4490-6/17/10 ...$15.00
https://doi.org/10.1145/3121050.3121098
Our long-term goal is to dynamically process user feedback for evaluating user reward during an interactive session and to respond accordingly as presented in Figure 1. For example, if a user clicks on images after issuing the query “panda,” an interactive system can infer a user’s reward to provide a better experience for the next step in the interaction.

The main aim of this paper is to study how user reward functions can be learned from their interactions, which we break down into the following concrete research questions.

**RQ1: How to define user reward?** We model the interactive user-system interface in Figure 1 using Reinforcement Learning (RL) [30]. More specifically, a user is an agent who interacts with an environment, which is an interactive system, in a sequential manner with discrete time steps, \( t = 0, 1, 2, 3, \ldots \). At each time step \( t \), the user receives some representation of the system, which we call *situation*, \( s_t \in S \), where \( S \) is the set of possible system situations, and on that the user performs an action, \( a_t \in A \), where \( A \) is the set of actions possible in the situation \( s_t \). We propose the following way of modeling the user reward function: one time step later, the user receives a numerical reward, \( R_{t+1} \in R \), and finds himself in a new situation, \( s_{t+1} \). While examining interactive user behavior, we should consider the reward function unknown and to be restored through empirical detection. This is specifically important for multiattribute reward functions as in our case. A unit of user interactions with the system within some time period \([t_i, t_i+j]\), called a *user session*.

**RQ2: How to recover a user reward function?** Our problem fits the setting of Inverse Reinforcement Learning (IRL) [25], which is defined as follows: given: (1) sequential users’ interactions with a system over time in a variety of circumstances, and (2) a system model; determine: the reward function of the users.

**RQ3: How to incorporate user features into the learning process?** IRL techniques have successfully been applied for the apprenticeship (or imitation) learning problem [25], to discover the reward function whose optimization would produce *desirable behavior*. One of the prominent examples is self-driving cars, where one recovers the reward function based on expert driving behavior. In contrast, we attempt to restore reward function(s) covering various types of user behavior (similar to driving styles in [1]). Similar to [4] we incorporate into our learning process user features that can be organized in two groups: (1) *static user features* remain unchanged during the user session, e.g., age, gender; (2) *dynamic user features* describe user behavior in the particular situation \( s_t \), e.g., time spent.

To confirm our hypothesis that different types of users have different reward functions, we perform a preliminary reward learning experiment for which we choose onsite logs of physical interactions in a museum because user features are explicitly given.

## 2 BACKGROUND AND RELATED WORK

**Evaluation.** Information Retrieval (IR) is about getting the right information to the right people in the right way. Evaluation has historically been one of IR’s key concerns. Offline, system-oriented evaluation, with a strong focus on assessing the degree to which a system is able to successfully identify documents that are relevant to a query, has received considerable attention [27]. In parallel, user-oriented evaluation methods for interactive information retrieval received considerable attention [16]. Increasingly, though, there is a realization that system aspects and user aspects should be assessed in tandem. In online experimentation the two aspects naturally come together [15].

Online controlled experiments, such as A/B testing or interleaving, have become widely used techniques for controlling and improving search quality based on data-driven decisions [19]. This methodology has been adopted widely [3, 7, 9, 31]. An A/B test is a between-subject test designed to compare two variants of a method (e.g., ranking on the SERP, ad ranking, colors and fonts of the web result title) at the same time by exposing them to two user groups and by measuring the difference between them in terms of a *key metric* (e.g., revenue, number of visits, etc.).

There are many existing studies towards better online evaluation that are devoted to improving the sensitivity of our measurement methods [28], inventing new metrics [8, 10] or improving existing ones [9]. An important goal of recent studies is to make metrics more consistent with long-term goals [19]. User engagement metrics show different aspects of user experience. For instance, they can reflect (1) *user loyalty* – the number of sessions per user [29]; (2) *user activity* – the number of visited web pages [21] or the absence time [10]. Periodicity engagement metrics of user behavior, which result from the discrete Fourier transform of state-of-the-art engagement measures, have also been proposed [8]. Few studies have looked at evaluating intelligent assistants in online settings [17, 18, 33, 34], where user satisfaction [16] is defined and predicted at the session level.

Most existing studies are directed towards generalization from user interaction rather than understanding the behavior of individuals. Very few works [2] have explored why users behave in particular ways by applying economic models.

**Reinforcement learning in interactive systems.** Several authors have adopted a Reinforcement Learning (RL) perspective on IR problems. Hofmann et al. [12, 14] seem to have been the first; they use RL for online evaluation and online learning to rank and define reward functions directly in terms of NDCG [13]. Later work on RL in IR predefined reward functions as the number of satisfied clicks in session search [22, 23]. Odijk et al. [26] use RL for query modeling and define reward in terms of retrieval performance (NDCG). Applications of IRL in interactive systems are relatively rare. Ziebart et al. [36] use IRL for predicting the desired target of a partial pointing motion in graphical user interfaces. Monfort et al. [24] use IRL to predict human motion when interacting with environment. It is not straightforward to apply IRL for interactive systems as it is often unclear how one should deal with user features [4].

To summarize, the key distinctions of our work compared to previous work are that we introduce a new problem (recovering user reward functions from interaction data) and propose a method to address the problem.

## 3 LEARNING USER REWARD FUNCTIONS

We start by investigating **RQ1**: How to define user reward? To model the user reward function, presented in Figure 1, we use a finite Markov Decision Process (MDP). An MDP is a tuple \((S, A, T, d_0, y, R)\), where \( S \) is a set of \( N \) states (possible system situations); \( A \) is a set of \( K \) actions; \( T \) is a set of state transition probabilities \( P_{SA}() \) is a state transition probability upon taking action \( a \) in state \( s \); the initial distribution of states is \( d_0 \) and \( y \in [0, 1) \) is a discount
factor; \( R \) is a user reward function, where \( R(s, a) \) is a reward given for action \( a \) in situation \( s \). Given a current state \( s \) and action \( a \) together with any next state \( s' \), the expected value of the next reward is: \( r(s, a, s') = E(R_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s') \).

An MDP without reward function is denoted as MDP/R, i.e., a tuple \((S, A, T, d_0, y)\). Let \( f : S \rightarrow [0, 1]^k \) be a vector of features over states. There is a “true” reward function \( R \) that is given by a linear combination of \( k \) features \( f_i \) with weights \( \theta_i \) where \( \theta^T \in \mathbb{R}^k \). We assume that the reward function is defined as \( R(s) = \theta^T \cdot f(s) \). In our setting, \( f \) is a vector of features describing user behavior, e.g., time a user spent interacting with the system state. The desired vector \( \theta^T \) specifies the relative weighting between these features.

A policy is a map \( \pi \) from situations, \( s_t \in S \), and actions, \( a_t \in A \), to the probability \( \pi(a_t \mid s_t) \) of taking action \( a_t \) when in state \( s_t \). The value function \( V \) for policy \( \pi \) is:

\[
E_{s_0 \sim d_0}[V(s_0)] = \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi \right] = \theta^T \cdot \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t f(s_t) \right] \mid \pi \quad (1)
\]

The expectation is taken with respect to random sequences of situations \( s_0, s_1, \ldots \) drawn from the starting situation \( s_0 \sim d_0 \). The goal of RL is to find \( \pi \) such that \( V^*(s) \) is maximized. There exists at least one optimal policy \( \pi^* \) such that \( V^*(s) \) is simultaneously maximized for all \( s_t \in S \) by \( \pi^* = \pi \) [25].

We need to estimate a user’s features expectations \( \mu_F \). Given a set of \( m \) user sessions \( \{s_0^i, s_1^i, \ldots, s_{T_i}^i\} \) generated by users, we empirically estimate \( \mu_F \) as \( \mu_F = \frac{1}{m} \cdot \sum_{i=0}^{m} \gamma^T f(s_i) \).

Next, we consider \( RQ2: \text{How to recover a user reward function?} \) The problem of IRL is to find a reward function that can explain the observed user behavior. We formulate the problem as follows: given (1) an MDP/R, (2) \( f \), (3) user feature expectations \( \mu_F \); determine a policy whose performance is close to the observed user group behavior based on the unknown user reward function \( R = \theta^T \cdot f \).

There are a number of available IRL methods [1, 6, 25, 37, 38]. For our preliminary experiments we adopt Maximum Entropy IRL (MaxEnt) [37] to recover the user reward functions.

4 INCORPORATING USER FEATURES

We report on a preliminary experiment aimed at assessing the feasibility of uncovering the reward function from interaction data.

Data. The dataset we use [11] is extracted from the physical interaction logs of an archaeological museum. Besides common exhibitions, this museum also provides additional information that can be obtained from different POIs. The contents at each POI are based on one specific topic and there are 8 topics in total. According to the corresponding topic, each POI shows 3 related objects and the objects at different POIs can be accessed in any order. Users can enter their personal information and preference at the beginning to personalize the contents being shown. The original dataset consists of 5 months of onsite logs with about 21,000 sessions. Each record contains one user’s personal information (e.g., age and language) and the interaction order for different objects at POIs. The starting time and how long the interaction lasts for each object are also recorded. After filtering the sessions which did not have any interactions or necessary user information (such as age), 4,694 out of 21,000 interaction sessions remain and constitute our final data.

Experimental design. To address \( RQ3: \text{How to incorporate user features into the learning process?} \) we propose the following experiment. First, we focus on exploring the difference between the reward functions of different groups. According to the user’s age, we divide the data into two groups, child and adult, with 1,135 and 3,559 sessions, respectively. To indicate each interaction situation, we consider three kinds of features: “topic” (8 different types, including: appearance, death, religion, architecture, entertainment, food, trade, army), “object order” (3 objects for each topic) and “duration time” (discretized in 3 bins denoting 0–30s, 30–90s and more than 90s).

One-hot encoding is used in our experiment as some features are categorical. In this manner, 14 features are selected and 72 situations are defined. With respect to the action feature, we use the object’s topic number to denote an action; taking a specific action means the user will transition to the situations that has the same topic number.

For each action feature, we define the probability \( \pi_{ab} \) to assign an action for each situation, empirically estimate \( \mu_F \) as \( \mu_F = \frac{1}{m} \cdot \sum_{i=0}^{m} \gamma^T f(s_i) \).

5 CONCLUSIONS AND FUTURE WORK

In this conceptual paper we investigated how reward functions can be directly learned from users’ interplays with an interactive system.

First, we explored \( RQ1: \text{How to define user reward?} \) We used a container of different features (such as system features and user features) to represent all possible situations. The user reward function, then, is a linear combination of situation features that we used to explain demonstrated user behavior.

Second, to answer \( RQ2: \text{How to recover a user reward function?} \) we proposed to use IRL techniques. We adopted maximum entropy IRL to recover the users reward functions. Our experimentation with a physical interaction dataset showed that the reward functions of different user groups have different priorities about features. Some features have a bigger impact on one user group’s reward than on another group’s.

Third, we studied \( RQ3: \text{How to incorporate user features into the learning process?} \) In the dataset used, explicit user features are given. We see two ways of incorporating user features into the learning process: (1) Grouping based on static user features; explicit user
features are needed and we recover the reward functions for all the groups separately; the drawback is that groups need to be predefined and the number of user features should be relatively small.

(2) Using user features to describe the situation. In many scenarios, such as web search, explicit user features may be unavailable and the first method is not applicable. How to adjust the reward for different users without predefined groups?

In conclusion, recovering user reward functions is (1) feasible, (2) a promising direction, and (3) applicable in many scenarios, including personal assistants, web search, recommender system.

As we are only at the beginning of our investigations into user reward functions, many questions remain open. (1) How can we make the reward function more complex, e.g., non-linear, rather than assuming that the function is a linear combination of situation features? (2) How can we make the system learn the rewards for different users automatically and return personalized rewards? To achieve this goal, user features should be taken into account during the learning process which can balance the reward for different users. Besides static user features, dynamic user features could also influence the decisions of users and need to be considered if possible. (3) How can we solve the computational problem of very large state spaces? The presence of very many features implies that many states will be defined, which in turn may reduce learning efficiency. Automatic feature construction and feature selection can be considered. (4) With more diverse and complicated features being considered, how can we adopt emerging techniques (such as deep inverse reinforcement learning) to mine user interaction scenarios effectively and efficiently?

Acknowledgments. This research was supported by Ahold Delhaize, Amsterdam Data Science, the Bloomberg Research Grant program, the Criteo Faculty Research Award program, the Dutch national program COMMIT, Elsevier, the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nr 612.001-116, HOR-11-10, CI-14-25, 652.002.001, 612.001.551, 652.001.003, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

Table 1: The recovered weights of reward function.

<table>
<thead>
<tr>
<th>Group</th>
<th>Appearance</th>
<th>death</th>
<th>religion</th>
<th>architecture</th>
<th>entertainment</th>
<th>food</th>
<th>trade</th>
<th>army</th>
<th>object1</th>
<th>object2</th>
<th>object3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adult</td>
<td>0.3378</td>
<td>0.1340</td>
<td>1.0539</td>
<td>-0.7106</td>
<td>0.9119</td>
<td>1.3577</td>
<td>0.4174</td>
<td>0.6783</td>
<td>0.6161</td>
<td>0.6556</td>
<td>0.6171</td>
</tr>
<tr>
<td>Child</td>
<td>0.9436</td>
<td>0.6479</td>
<td>1.2215</td>
<td>0.2658</td>
<td>1.2394</td>
<td>1.5001</td>
<td>0.7700</td>
<td>0.4649</td>
<td>0.9405</td>
<td>0.7115</td>
<td>0.7368</td>
</tr>
</tbody>
</table>