Learning User Preferences by Observing User-Items Interactions in an IoT Augmented Space

David Massimo
Free University of Bolzano-Bozen
Piazza Domenicani, 3
Bolzano, Italy
damassimo@inf.unibz.it

Mehdi Elahi
Free University of Bolzano-Bozen
Piazza Domenicani, 3
Bolzano, Italy
meelahi@unibz.it

Francesco Ricci
Free University of Bolzano-Bozen
Piazza Domenicani, 3
Bolzano, Italy
fricci@unibz.it

ABSTRACT

Recommender systems generate recommendations by analysing which items the user consumes or likes. Moreover, in many scenarios, e.g., when a user is visiting an exhibition or a city, users are faced with a sequence of decisions, and the recommender should therefore suggest, at each decision step, a set of viable recommendations (attractions). In these scenarios the order and the context of the past user choices is a valuable source of data, and the recommender has to effectively exploit this information for understanding the user preferences in order to recommend compelling items.

For addressing these scenarios, this paper proposes a novel preference learning model that takes into account the sequential nature of item consumption. The model is based on Inverse Reinforcement Learning, which enables to exploit observations of users' behaviours, when they are making decisions and taking actions, i.e., choosing the items to consume. The results of a proof of concept experiment show that the proposed model can effectively capture the user preferences, the rationale of users decision making process when consuming items in a sequential manner, and can replicate the observed user behaviours.

1 INTRODUCTION

Recommender Systems (RSs) are software tools supporting users' decision making processes. They learn users' preferences and offer to them personalised suggestions for items to consume or actions to perform [15]. RSs identify recommendable items for a target user on the base of her explicit preferences (i.e., ratings or likes) or by leveraging the hidden preference information extracted from user actions while interacting with items, such as, clicking on pages describing these items [5, 7].

However, those items the user explicitly ignores, or the specific order in which the user acts on them, is a valuable information that has not been fully exploited yet. For instance, Collaborative Filtering [7] and Content-Based [9] techniques use only the unordered set of items that the user liked or on which the user expressed an implicit feedback (click). Besides these models, other techniques, which take into consideration the sequential aspect of item consumption, have been developed. While Sequence Mining (SM) tries to discovery item consumption patterns [6, 12], other techniques adapt the system behaviour to provide sequences of items to be consumed by the user [11, 13, 17]. However, SM techniques do not learn the user's rationale for consuming items sequentially. In particular, they do not generate explicit knowledge about how a user decides to consume certain items rather than others or why she chooses items in a specific order. The approaches in the second group, instead, focus on how to adapt to the system behaviour to promote the consumption of specific items in a sequence, without modelling the user behaviour when consuming items in a sequence.

In this work, we address these limitations by proposing a user modeling approach that learns user's preferences in sequential decision making scenarios, and can be used to suggest items to be consumed in a sequence. Predicting (or recommending) a full sequence of items is appealing in scenarios where the user is actually faced with multiple choices, e.g., which museum's exhibit to visit next. User preference learning is made possible by (1) exploiting observations of user's actions while the user is consuming items, and (2) leveraging item's descriptive features and environment's contextual information.

In order to reach our goal, we propose an approach based on Inverse Reinforcement Learning (IRL) [16]. IRL allows the RS to learn a policy that dictates for each possible state the best action the user should perform (which item to consume). This policy fully describes the sequential preferences of a user or a group of users and can generate relevant recommendations for these users, as well as, new users deemed as similar to the observed ones. For instance, when facing a new user, the system can reuse a model learnt for the users who have a similar information need (goal) or belong to the same socio-demographic group. IRL accomplishes user's preference learning by discovering the importance weights of features of items at disposal for being consumed (more precisely, features describing the state of the user). These weights define the utility (reward) a user gains by consuming items.

IRL is a promising technology that can be adopted in variety of application scenarios. However, in this work, we mainly focus on scenarios where users are faced with a sequence of choices. Indeed, we apply this technology to deal with user preference elicitation and modeling by relying on user behaviour observations that are obtained from a network of Internet of Things [8]. In this scenario, the items to recommend are active and can signal their identity and proactively interact with the users. IoT technologies open the

UMAP'17 Adjunct, July 09-12, 2017, Bratislava, Slovakia

978-1-4503-5067-9/17/07…$15.00
DOI: 10.1145/3099023.3099070

way to new types of interactions between users, items and the recommender.

As a first case study, this work focuses on learning the user model for a recommender that suggests multimedia items, which illustrates exhibits, in a Museum. In particular, we study how visitors' preferences can be learnt, by relying on the observation of the users' consumption of the available media. The experiment results show that, by just using low-level behaviour data, our approach can learn users' preferences (reward function) and the policy adopted by users when consuming items sequentially.

The rest of the paper is organised as following. Section 2 reviews the related work. In section 3, we describe the technical details of the proposed model and in section 4, we present a case study. In section 5 we comment the observed results, and, in section 6 we conclude the paper with a discussion of the results and future work.

2 RELATED WORK

In the RSs literature some works studied the sequential aspect of item consumption, mainly by mining patterns of user behaviour, when browsing the web [12] or listening to music [6]. However, these techniques, which try to discover hidden patterns in the data, are rather generic predictive methods and were not specifically developed for modelling sequential decision making process. This means to understand what properties of the items are connected to the choices of the users and therefore to capture "why" users are behaving in a particular way. The main limitation of pattern-discovery approaches relies on the fact that the sequences of the training items are composed only of items that have been already consumed by users. Even though these approaches learn the frequency of consumed items within a session (e.g., a time window shopping) and/or the similarities of consumed items (e.g., product categories) they are not meant to discover the rationale of users in consuming items in a certain order.

Other related approaches adopt a different perspective and instead of modelling user's decision making, they try to optimize the performance of the recommender while interacting with the user. They identify the optimal sequence of actions or recommendations that the system should provide to help the user to make a choice, or a sequence of choices, and to accept the system recommendations [11, 13, 17]. The considered sequential decision making and recommendations task is modelled as a Markov Decision Process (MDP), and Reinforcement Learning (RL) is used to learn which action the system must perform in each state it may be while interacting with a user (optimal policy). Indeed, from the RSs perspective, RL is a natural approach to optimize the system behaviour when acting in an environment which is defined by the user reactions to the system actions [11, 13, 17].

When modelling the system behaviour with MDP and RL one must conjecture the reward/utility that the system obtains when an action is performed. For instance, if the action is the recommendation of an item, a positive reward should be obtained if the item is accepted by the user and no reward, otherwise. When instead the system is modeling user preferences and behaviour, the reward obtained by the user when performing an action is unknown. In this case, the goal is to infer the user's preferences from the observed user's behaviour, and the reward that the user obtains from performing actions (e.g., consuming an item) is not easy to estimate. In fact, the user rarely evaluates the outcome of a recommendation or the benefit obtained from consuming an item. In other words, when modeling the user decision maker with an MDP, the problem of learning user's preferences remains open. Hence, the above mentioned applications of MDP and RL to RSs do not contribute to the real understanding of the individual and collective user's preferences, especially when users are consuming items sequentially.

In addressing that limitation, in this paper, we propose a novel approach that primarily allows the RS to learn the utility that the users obtain from the performed actions by simply observing their behaviours and assuming that they act optimally with respect to their utility function. Hence, in the preference elicitation process, one must focus on users considered to be "experts" in the domain. This problem is known as Apprenticeship Learning and can be addressed with a machine learning technique called Inverse Reinforcement Learning (IRL)[16]. IRL exploits observations of agents, which are considered to be expert, i.e., they act optimally, in order to infer the reward function that scores the value of each action performed by the agent[1]. From this perspective, IRL task is closer to that of RSs based on implicit feedback, which exploit data that signals only indirectly users' preferences or opinions [5, 7]. These RSs predict whether the user will act on a target item and interpret this prediction as a discovered preference for the item. In our setting, by applying IRL, we exploit the same idea, but we learn user's preferences in a sequential decision making scenario rather than in isolated decision making tasks.

In this work, we adopt IRL to deal with preference elicitation of the users in complex decision making scenarios, and the collection of behavioural observations is obtained by exploiting Internet of Things technologies. Internet of Things (IoT) is a dynamic and global adaptive network assisted by an intelligent coordination of the communication between the connected devices, the so called *things* [8]. Technologies, protocols and architectures devised for the IoT has been summarized in [8]. IoT relies on the following technologies: Radio-Frequency IDentification (RFID), Near Field Communication (NFC) and Wireless Sensor Networks (WSN) which are networks that connect these sensors via wireless communication. Due to the huge variety of sensors and application scenarios, IoT networks are a complex playground in which to experiment user's preference learning. We focus on real world applications in tourism and cultural heritage [3, 14], where IoT registers the interactions between visitors at an art exhibition and interactive media station (deployed as WSNs) spread over the exhibition area.

3 TECHNICAL APPROACH

We model the user decision making problem as a finite Markov Decision Process (MDP), i.e., a tuple (S, A, T, r, γ), in which S is a finite set of states, A is a finite set of actions, T is a finite set of transition probabilities, $r : S \rightarrow \mathbb{R}$ is a reward function and $\gamma \in [0, 1)$ is a discount factor. When an agent/user in a state $s_0 \in S$ performs action $a \in A$, it earns an immediate reward $r(s_0)$ and moves, with probability $T(s_0, a, s)$, to a new state $s \in S$. For an MDP we define $\pi : S \longrightarrow A$ to be a policy, i.e., a function that maps states to actions, and gives the action, to be performed, for each state. If a policy maximises expected utility (reward) it is said to be optimal and is denoted as π^*.

Given an MDP, the value-iteration algorithm can be used to compute action-value function $Q : S \times A \longrightarrow \mathbb{R}$. The action-value function for a state s and action a provides an estimate of the discounted total reward, which is obtained by the agent, when taking action a in state s. Below we show the Q function for the i-th iteration cycle of the learning algorithm:

$$Q_i(s,a) = \sum_{s'} T(s,a,s')r(s') + \gamma \max_b Q_{i-1}(s',b) \tag{1}$$

This is also known as Bellman equation and characterizes the optimal behaviour of the agent.

3.1 Reinforcement Learning

An agent task is finite when it is completed after a fixed number of steps. The number of time steps an agent plans for its task is called horizon H of the MDP. A task is defined by a sequence of decision rules, one for each time step in the planning horizon. The decision taken by an agent are represented by a policy π, which is called Markovian if for each time step t in the planning horizon the decision rule $\pi_t : S \longrightarrow A$ depends only on the current state s_t. When the transition probability of moving from one state s_0 to a state s by choosing action a is given by $T(s_0, a, s) \in \{0, 1\}$, the MDP is called deterministic. In a deterministic MDP a policy is equivalent to a sequence of H actions for each possible initial state. The value of a policy π from an initial state $s_0 \in S$ is defined as $v_\pi(s_0) = E_\pi\big[\sum_{t=0}^{H-1} \gamma r(s_t)\big]$. This is the expected value of the discounted total reward the agent obtains by taking the action at time t according to π. The reinforcement learning goal, given an MDP, is to compute a policy π that maximizes the value function $v_\pi(s_0)$.

3.2 Inverse Reinforcement Learning

Inverse reinforcement learning exploits observations of an agent actions and identifies an MDP$\backslash r$, which is a tuple (S, A, T, γ), that is consistent with the observed behaviours. The observations are assumed to come from rational one or more agents that express a behaviour at the state-action level. More formally, given a state space S, an action set A, transition probabilities T, the goal is to find a reward function r which rationalizes the observed behaviour of the agent. In other words, the reward represents agent's preferences over states of a specific MDP. For finite state spaces, the reward function can be represented as a vector of real numbers $r \in \mathbb{R}^{|S|}$, where each component gives the reward for one state. The reward r can also be assumed to be a linear function of a feature-based representation of the state:

$$r(s) = \theta \cdot \phi(s) \tag{2}$$

Here, $\phi(s)$ is a n-dimensional feature vector representation of the state $s \in S$, and $\theta \in \mathbb{R}^n$ is an unknown weight vector modelling the user "preference" for the state features. In this case, the IRL goal is to learn the vector θ in order to compute the reward r. Since in many real cases the agent policy π is unknown, the goal of IRL is to infer it from observations of experts agents (those acting rationally). Observations are sequences of states in S and actions in A that represents the transitions of the agent in the MDP. From these information both the reward r and the policy π can be learnt. In particular, let us assume that we have observed N finite sequences of state-action pairs $O = \{\zeta_1, \dots, \zeta_N\}$ that are made by the agent. We assume all the trajectories have length H, and a trajectory ζ_j is of the form $\zeta_j = \big((s_{j_0}, a_{j_0}), (s_{j_1}, a_{j_1}), \dots, (s_{j_{H-1}}, a_{j_{H-1}})\big)$ where $s_t \in S$ and $a_t \in A$. In case all the observed trajectories derive from a policy π^* and the agents act optimally by following this policy, then we can assume that the policy is optimal with respect to the reward function r.

As pointed out in [1], the problem to identify a reward function, whose corresponding optimal policy describes correctly the agents' behaviour, is ill-posed. In fact, the problem is under-determined: for instance, if the reward function $r(\cdot) \equiv 0$, the optimality equation is satisfied by any policy π^*. This problem can be solved by making further assumptions on the properties of the optimal policy. For instance, one can search a solution that maximises the difference between the optimal reward and the reward derived from alternative policies [1].

In order to understand user's preferences we focus on a model that is structural, i.e., it explains the preferences in decision-making task of an agent, by taking a probabilistic approach. In particular, we assume that the observations O are samples from a family of probability distributions, which depend on the unknown reward function, allowing for suboptimal behaviour. In this setting, we model agent's choice by means of the Boltzmann distribution for action selection, which induces variability in the behaviour that is tied to the values of the actions. This distribution is also used as a model for human decision making [10] and is exploited in the IRL technique called Maximum Likelihood Inverse Reinforcement Learning [2], which is the approach we have used in this study. We would like to point out that the variability induced by the Boltzmann distribution for action selection $\pi_\theta(s,a) = e^{\beta Q(s,a)} / \sum_{a'} e^{\beta Q(s,a')}$ promotes the exploration of the states in S of the MDP, hence it allows to build generative models that can be used to simulate behaviours under new circumstances.

3.3 Maximum Likelihood IRL

Maximum Likelihood Inverse Reinforcement Learning (MLIRL) assumes that experts randomize individual action choices [2]. Choice actions are sought by a maximum likelihood solution via gradient ascent. MLIRL exploits the fact that a guessed reward function θ induces a probability distribution over the action choices and hence determines a likelihood for the observations in S. Expected values (discounted) are computed via the following formula:

$$Q_\theta(s,a) = \theta^T \phi(s,a) + \gamma \sum_{s'} T(s,a,s') \frac{\sum_a Q(s,a) e^{\beta Q(s,a)}}{\sum_{a'} e^{\beta Q(s,a')}} \tag{3}$$

The formula is a variation of the Bellman equation in 1 that replaces the maximisation term with values composition via Boltzmann exploration. This approach makes the likelihood (infinitely) differentiable. By exploiting Boltzmann exploration policy the log likelihood of the observations is:

$$L(D|\theta) = \log \prod_{i=1}^{N} \prod_{(s,a)\in\zeta_i} \pi_\theta(s,a) = \sum_{i=1}^{N} \sum_{(s,a)\in\zeta_i} \log \pi_\theta(s,a) \qquad (4)$$

MLIRL looks for $\theta = \arg\max_\theta L(D|\theta)$ which is the maximum likelihood solution that is found via gradient ascent optimisation. The solution is known to converge in finite-horizon settings. The existence of multiple reward functions for which an observed trajectory is optimal in a given MDP, is solved by assigning high probabilities to observed behaviour and low probability to the unobserved. MLIRL general steps are illustrated in Algorithm 1.

Algorithm 1 Maximum Likelihood Inverese Reinforcement Learning (MLIRL)

```
Input: MDP $\backslash r$, state features $\phi$, observations $\{\zeta_1, \ldots, \zeta_N\}$, number of steps M, $\lambda_t$ step size.
$\theta \leftarrow$ Initialize with random values;
for t=1 to M do
    Compute $Q_{\theta_t}$,$\pi_{\theta_t}$
    $L = \sum_i \Pr(\zeta_i) \sum_{(s,a)\in\zeta_i} \log \pi_{\theta_t}(s,a)$
    $\theta \leftarrow \theta + \lambda_t \nabla L$
end for
Output: learnt $\theta$
```

Once the reward function r and the optimal policy π^* are known, the system is also able to generate recommendations. That is, for those users whose profiles (the observed sequences of state-action pairs ζ_i) that have been already learnt, the known policy could be exploited to provide them with suggestions of actions to perform. This could be done even in another MDP with the same state features of the MDP on which the learning is performed. In addition, sequences of items could be recommended by clustering the sequences ζ_i observed in different MDPs. This is done in order to identify common intents and the new policies. By leveraging those policies and rewards new sequences of items can be suggested. Learnt policies can be used to provide to new users suggestions in the form of action sequences in the MDP. Moreover, the vector of preferences θ could be exploited to provide recommendations about items in domains that differ from the observed one by exploiting traditional RS techniques.

We would like to point out that the proposed approach for preference learning and recommendations can be seen as an hybrid technique that exploits implicit feedback from users and item features to learn user's behaviour in consuming items sequentially. However, even though the reward vector θ could be learnt by means of a standard content based approach, it would not be able to produce a meaningful sequence of items for a target user. Besides, as we already mentioned, sequence mining techniques would not provide any rationale for users sequential decision making behaviour.

4 CASE STUDY

4.1 Dataset

In our experiments, we used a dataset containing logs of user-media interactions, which was collected during an exhibition called *Voices from the past in fort Pozzacchio* [14], which installation was developed within the meSch European project.

The exhibition was focused on World War I, and was hosted by *Museo della Guerra* (Rovereto, Italy). Here, IoT devices were used to allow the visitors to interact with a number of media stations [1] that offer multimedia items (video and audio). Each of the stations were equipped with a RaspberryPi board, connected to NFC reader(s) and either a video projector or audio speakers. Each visitor was provided with a NFC tag, placed into a "pebble". Visitors could activate media playing at the stations by simply placing their "pebbles" near one of the NFC readers. Each media station was carefully designed to cover a certain topic of the war, that reflects the narration perspective of the multimedia item. The shortest media lasts circa one minute, whereas the longest is over two minutes. Table 1 presents some information about these media items.

The exhibition area in the museum was composed of four spaces: entrance, three showrooms. In the entrance space, 3 video items (1 to 3) could be played, introducing the tour and presenting the history of fort Pozzacchio (topics *intro* and *fort*), whereas those, which could be watched in the first exhibition area were 4 graphic animations, that we consider as video (4 to 8), on the topic of *buildings*. 5 audio items (9 to 13) on the the topic of *civilians* were provided in the second exhibition area. In the last space, 4 video items (14 to 17) on topic *soldiers' life* were shown.

The collected data logs contain the sequence of the timestamped interactions, made by the visitors with the media stations. Each log entry contains the unique identifier of the visitor (stored in the NFC tag) and the played multimedia item *id*. This data combined with information about the media (features) allows to understand user's movements within the exhibit area and whether a multimedia item was played or not. We combined the log data with observations made by staff-members during the visit session of a sample of 40 users. These observations, which after filtering are in the number of 24, indicate whether visitors consumed completely, partially or not a media.

4.2 State Space

We denote with $A = \{c_j : j = 1, 2, 3\}$ the set of possible actions of consuming a multimedia item: consume, consume partially, and skip, respectively. We consider the following state model: $S = \{(m_i, c_j) : i = 1, \ldots, 17, j = 1, 2, 3\}$. The pairs (m_i, c_j) represent the media m_i consumed by action c_j. Out of the 3 possible actions and the 17 multimedia items we derived 51 possible states that describe the consumption behaviour of a visitor for the items.

The following features were used for representing the states: **media perspective** $f_p \in \{0, 1\}$, where $p \in \{intro, fort\_today, building, civilians, soldiers\}$; **media duration** $f_d \in \mathbb{R}$; **media type** $f_i \in \{0, 1\}$, where $i \in \{audio, video\}$ and **consumption status** $f_c \in \{0, 1\}$, where $c \in \{consume, partial, skip\}$. More state features could have been used but we preferred here a simple approach to better test the model learning procedure.

4.3 MDP Model

The group of visitors (one or more persons) is modelled as an individual agent who is autonomously taking decisions in order

[1] https://www.youtube.com/watch?v=DReu2J7eWx4&t=41s

Table 1: Information about media.

Media	Perspective	Duration	Type
1	Intro	65	video
2	Fort Today	65	video
3	Fort Today	52	video
4	Building	81	video
5	Building	57	video
6	Building	49	video
7	Building	64	video
8	Building	54	video
9	Civilians	55	audio
10	Civilians	84	audio
11	Civilians	122	audio
12	Civilians	72	audio
13	Civilians	61	audio
14	Soldiers	145	video
15	Soldiers	119	video
16	Soldiers	141	video
17	Soldiers	104	video

to optimize an unknown utility function or reward r. This utility, could be intuitively seen as the fulfilment of the cultural interest of the visitor at the museum while discovering its collection.

Visitors are assumed to plan the visit of the reachable items in a session. The time horizon of the users' visiting session is $H = 17$, and it corresponds to the number of transitions through the media stations available at the exhibition. The set of actions A, a user can perform and the state space S have been introduced in 4.2. Here we detail the possible state transitions. Being in the state $s = (m_l, c_k)$ means having performed action c_k on media m_a, which in the horizon H means performing action c_k at time l. The transition probabilities of the MDP are deterministic:

$$T\big((m_l, c_k), c_s, (m_i, c_j)\big) = \begin{cases} 1 & \iff i = l + 1 \text{ and } c_j = c_s \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Our goal is to learn the vector θ, which define the reward function, and in order to do that, we consider observations of expert visitors in the form of trajectories defined by the sequence of state-action pairs. The observations are collected by mining the data log discussed in section 4.1, represented as sequences of states (see section 4.2). By using MLIRL, we compute the (linear) reward function of the states. We denote with $O = \{\zeta_1, \zeta_2, \dots, \zeta_N\}$ the set of observed trajectories. A trajectory $\zeta_i = \big((m_{i_0}, c_{i_0}), \dots, (m_{i_{H-1}}, c_{i_{H-1}})\big)$ is composed by a sequence of H states that represent the sequence of consumption action c_{i_j} of media m_{i_j} during a visit session. For example, a visitor v trajectory is
$\zeta_v = \big((m_1, c_2), (m_2, c_3), (m_3, c_3), (m_4, c_1), \dots, (m_{17}, c_1)\big)$.

The visitor v starts by partially consuming (action c_2) the first media item (m_1), which leads to the second media (m_2) which he skips (action c_3). Then, the visitor skips also the successive media (third media m_3), action that leads to the following media (m_4) that is consumed (action c_1). The last action that the visitor performs is to consume the last media (m_{17}, c_1).

In our experiments, we set the discount factor to $\gamma = 0.9$, which decreases the value of the future rewards, and the probability to make a random transition to 20%. The other parameters of the MLIRL algorithm that we used are $\beta = 0.75$ as suggested in [2], which promotes exploration during the learning of the reward, and step size $\lambda_t = \frac{1}{\sqrt{t}}$ as suggested in [4]. The termination condition is given by the (L_2) norm $\|\theta_t - \theta_{t-1}\|_2 < 0.01$ of the vector θ computed over the $M_t = 500$ iterations.

Table 2: Learnt θ, vector of user preferences.

Feature	Value	Feature	Value
status partial	0.66	perspective Intro	1.97
status complete	3.13	perspective Fort	1.74
status skip	0.61	perspective Building	1.92
media type audio	1.99	perspective Civilians	2.07
media type video	1.46	perspective Soldiers	2.61
media duration	2.26		

5 RESULTS

Table 2 shows the vector of weights θ that was learnt by MLIRL by considering the available observations. The learnt values are here increased by a constant ($z = 2.45$) to make them all positive and ease their analysis. This vector defines the reward function and shows the feature preferences of the observed experts. The results indicate a correspondence between the learnt preferences and the user behaviours. Firstly, those states in which the multimedia item is consumed (those containing the feature *status complete*) have higher weight (3.13) than the states in which the content is partially consumed or skipped (0.66 and 0.61, respectively). Then we can observe the users' preferences for topics. We can see that the narration perspective *Soldiers* has a value of 2.61, which places it as the most liked topic. The other topics values are 2.07 for *Civilians* and 1.92 for *Building*, whereas narration perspectives *Fort* and *Intro* have values 1.74 and 1.97.

To show that the model is both generative, i.e., it can be used for simulations, and it is structural, i.e., it captures the rationality of the experts (their preferences), we show here the results of two additional experiments.

The results of the first experiment is shown in table 3. It is a comparison of the frequencies of the observed actions, for some examples of multimedia item, when the original 24 trajectories of experts are considered vs. when two samples of again 24 synthetic trajectories are generated by using the policy learnt by training the model with only 5 or 10 real observed trajectories. We observe a good match between the simulated and the observed data.

The second experiment aims at comparing the preference model learnt by using the real 24 trajectories of experts with the models learnt with the aforementioned synthetic trajectories. Figure 1 illustrates the average reward computed by policies obtained by learning the model with an increasing number of trajectories (from 1 to 24) which are either the observed ones or simulated by exploiting the policies learnt by training the model with only 5, 10, or 15 real trajectories. We assume that the correct reward function (and the true optimal policy) is obtained by training the model with the whole set of real observations. These results show that by using synthetic trajectories, the learnt policies performance scores tend to approximate those obtained by using all the observations of the experts, as more training trajectories are used.

Table 3: Comparison of observed and generated trajectories.

	Obs.			Gen. (5)			Gen. (10)		
media	c_1	c_2	c_3	c_1	c_2	c_3	c_1	c_2	c_3
2	19	5	3	16	0	2	20	0	1
4	21	3	2	18	4	3	19	3	1
6	24	1	0	19	1	3	21	0	1

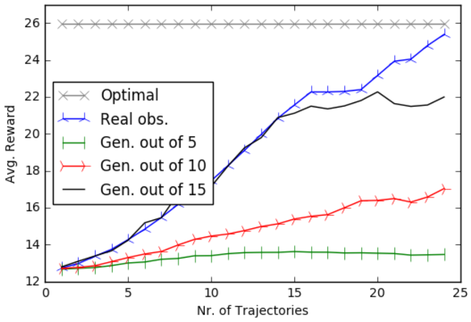

Figure 1: Plot of average reward computed for different numbers of real and simulated trajectories.

Overall, these results are promising and they offer an initial evidence of the effectiveness of the proposed IRL approach in preference learning elicitation and recommendation generation, even with such a small amount of data. Indeed, it shows how it can effectively tackle cold-start scenarios that traditional approaches may fail to cope with.

6 CONCLUSION AND FUTURE WORK

In this paper we have addressed the challenging problem of learning user preferences from low-level behaviour data by modelling and learning them in a sequential decision making problem. By exploiting the users' policy learnt by leveraging observations of their interactions with items, we could provide to the users suggestions about items to consume or actions to perform in different environments, whereas unknown users can be provided with suggestions for items/actions sequences that follow the learnt policy. We propose here a novel approach that is based on a machine learning technique called Inverse Reinforcement Learning.

We have performed some preliminary experiments by using a dataset, collected during an exhibition in a museum, which contains the log of user interaction with media stations (disposed on a fixed sequence), that offer to the users multimedia items related to World War I and allow the users to choose which items to play. The results of the experiments show that, by adopting this user modelling approach one can successfully learn the preferences of the users, which are here modelled with a reward function, which can be used then for recommendation generation. The proposed technique was exploited in an IoT scenario. But, other solutions for collecting user interactions are usable as well (e.g., in a web application by logging user actions).

The main limitation of this work is the size of the dataset that has been used for the experiments. The small number of multimedia items produced a relatively small state space. In the future, we would like to investigate the scalability of this approach when the number of items grows, which will result in a substantial growth in the number of states. Moreover, we plan to conduct additional experiments in order to compare the predictive capabilities of the devised solution with the other state-of-the-art approaches.

7 ACKNOWLEDGEMENTS

The research described in this paper is supported by the project Suggesto Market Space, funded by the Autonomous Province of Trento with Ectrl Solutions and Fondazione Bruno Kessler.

REFERENCES

[1] A. Y., Ng and S. Russell. 2000. Algorithms for inverse reinforcement learning. In *17th Int. Conf. on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 663–670. arXiv:arXiv:1011.1669v3

[2] M. Babes-Vroman, V. Marivate, K. Subramanian, and M. Littman. 2011. Apprenticeship learning about multiple intentions. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011* (2011), 897–904.

[3] F. Bohnert and I. Zukerman. 2014. Personalised viewing-time prediction in museums. *User Modeling and User-Adapted Interaction* 24, 4 (2014), 263–314.

[4] S. Ermon, Y. Xue, R. Toth, B. Dilkina, R. Bernstein, T. Damoulas, P. Clark, S. DeGloria, A. Mude, C. Barrett, and C. P. Gomes. 2015. Learning Large Scale Dynamic Discrete Choice Models of Spatio-Temporal Preferences with Application to Migratory Pastoralism in East Africa. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Pattern* (2015), 644–650.

[5] T. Gurbanov, F. Ricci, and M. Ploner. 2016. Modeling and Predicting User Actions in Recommender Systems. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization (UMAP '16)*. ACM, New York, NY, USA, 151–155.

[6] N. Hariri, B. Mobasher, and R. Burke. 2012. Context-aware Music Recommendation Based on Latenttopic Sequential Patterns. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. ACM, New York, NY, USA, 131–138.

[7] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback. *IEEE International Conference on Data Mining* (2008), 263–272.

[8] S. Li, L. D. Xu, and S. Zhao. 2015. The internet of things: a survey. *Information Systems Frontiers* 17, 2 (2015), 243–259. arXiv:arXiv:1011.1669v3

[9] P. Lops, M. De Gemmis, and G. Semeraro. 2011. *Content-based recommender systems: State of the art and trends*. Springer US, Boston, MA, 73–105.

[10] R. D. Luce. 1959. *Individual Choice Behavior: A theoretical analysis*. Wiley.

[11] T. Mahmood, F. Ricci, and A. Venturini. 2009. Improving Recommendation Effectiveness: Adapting a Dialogue Strategy in Online Travel Planning. *Information Technology & Tourism* 11, 4 (2009), 285–302.

[12] B. Mobasher, H. Dao, T. Luo, and M. Nakagawa. 2002. Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks. *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* (2002), 669–672.

[13] O. Moling, L. Baltrunas, and F. Ricci. 2012. Optimal radio channel recommendations with explicit and implicit feedback. *Proceedings of the 6th ACM conference on Recommender systems - RecSys '12* (2012), 75.

[14] D. Petrelli, L. Ciolfi, D. van Dijk, E. Hornecker, E. Not, and A. Schmidt. 2013. Integrating Material and Digital: A New Way for Cultural Heritage. *interactions* 20, 4 (July 2013).

[15] F. Ricci, L. Rokach, and B. Shapira. 2015. *Recommender Systems: Introduction and Challenges*. Springer US, Boston, MA, 1–34.

[16] S. Russell. 1998. Learning agents for uncertain environments (extended abstract). *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)* (1998), 101–103.

[17] G. Shani, D. Heckerman, and R. I. Brafman. 2005. An mdp-based recommender system. *Journal of Machine Learning Research* (2005), 1265–1295.