Popularity, novelty and relevance in point of interest recommendation: an experimental analysis

David Massimo¹ · Francesco Ricci¹

Received: 17 May 2021 / Revised: 3 September 2021 / Accepted: 25 September 2021 /
Published online: 21 October 2021
© The Author(s) 2021

Abstract

Recommender Systems (RSs) are often assessed in off-line settings by measuring the system precision in predicting the observed user's ratings or choices. But, when a *precise* RS is on-line, the generated recommendations can be perceived as marginally useful because lacking novelty. The underlying problem is that it is hard to build an RS that can correctly generalise, from the analysis of user's observed behaviour, and can identify the essential characteristics of novel and yet relevant recommendations. In this paper we address the above mentioned issue by considering four RSs that try to excel on different target criteria: precision, relevance and novelty. Two state of the art RSs called SKNN and s-SKNN follow a classical Nearest Neighbour approach, while the other two, Q-BASE and Q-POP PUSH are based on Inverse Reinforcement Learning. SKNN and s-SKNN optimise precision, Q-BASE tries to identify the characteristics of POIs that make them relevant, and Q-POP PUSH, a novel RS here introduced, is similar to Q-BASE but it also tries to recommend popular POIs. In an off-line experiment we discover that the recommendations produced by SKNN and s-SKNN optimise precision essentially by recommending quite popular POIs. Q-POP PUSH can be tuned to achieve a desired level of precision at the cost of losing part of the best capability of Q-BASE to generate novel and yet relevant recommendations. In the on-line study we discover that the recommendations of SKNN and Q-POP PUSH are liked more than those produced by Q-BASE. The rationale of that was found in the large percentage of novel recommendations produced by Q-BASE, which are difficult to appreciate. However, Q-BASE excels in recommending items that are both novel and liked by the users.

Keywords Context-aware Recommender Systems · User behaviour learning · Sequential decision making · Clustering · Inverse reinforcement learning · Recommender System evaluation · User studies · Tourism Recommender Systems

✉ David Massimo
davmassimo@unibz.it

Extended author information available on the last page of the article

1 Introduction

Nowadays, tourists are accessing a wide spectrum of on-line information services that enable them to search and discover various types of travel-related services, such as, city tours, accommodations, food services and many others (World Tourism Organization 1995; Revfine.com 2019). This huge variety of tourism-related items, which are searchable and potentially findable on the web, have created a vast virtual landscape of possibilities for tourists and service providers. In fact, in the attempt to ease tourists' items search and selection processes and increase the number of searches that result in real bookings and purchases, the most prominent players of the tourism industry have introduced various types of Recommender Systems. As a matter of fact, key players, such as, Expedia,¹ Booking² and Kayak³ have developed business models rooted in information search and recommendation technologies.

More in general, Recommender Systems (RSs) are software tools that aim at easing human decision making in on-line information search and discovery scenarios (Ricci et al. 2015). In tourism applications, recommendation processes are characterised by specific facets. Firstly, tourists often search information in different contextual situations and for a wide range of activities that vary in nature and extent. For example, a tourist may prefer to visit a museum on a hot and humid day, but she may choose a park when the weather is mild. Context-Aware RSs have been developed to intelligently adapt the recommendations to the tourist's context (Adomavicius and Tuzhilin 2015). Besides, tourists typically search and consume more than a single service and perform more than one activity in their visit to a destination. In other words, they follow visit trajectories or itineraries, which are composed by successions of points of interest (POIs). Therefore, session- and sequence-aware RSs have been introduced (Mobasher et al. 2002; Jannach et al. 2017). The next-POI recommendation problem has been defined as correctly suggesting POIs that the tourist may be interested to visit next, i.e., after she has visited already some other POIs during the same day or in a previous visit (Cheng et al. 2013; He et al. 2016). We note that while general recommendation tasks have been addressed by the tourism industry, such as, recommending a city destination or a hotel, technical solutions of this more complex next-POI recommendation problem have not been yet implemented and deployed.

In Massimo and Ricci (2018, 2019) we have developed a novel next-POI context-aware RS that we call here $\mathcal{Q}$ -BASE. It is designed to help tourists to choose POIs, one after the other, during their visits to a city. $\mathcal{Q}$ -BASE models with a reward function how a POI is estimated to be relevant for a user and recommends POI-visits that are estimated to give a large reward. Moreover, we observe that users typically provide scarce and not explicit feedback (e.g., ratings) and are not keen to disclose behavioural data (Smith et al. 1996; Perentis et al. 2015). Hence, $\mathcal{Q}$ -BASE learns the reward function by applying Inverse Reinforcement Learning (IRL) (Ng and Russell

¹ <http://www.expedia.com>.

² <http://www.booking.com>.

³ <http://www.kayak.com>.

2000), which does not require explicit users' feedback and can learn the reward function even when a few user's behaviour observations are available, by making the simplifying assumption that the reward is the same for the tourists in a cluster of tourists making similar visit trajectories.

In Massimo and Ricci (2018) a preliminary off-line experiment showed that a state of the art nearest neighbour RS, namely SKNN (Jannach et al. 2017), generates more precise recommendations compared to Q-BASE. This means that SKNN more often includes in the recommendation list the precise next POI-visit performed by the tourist. But, it was also shown that Q-BASE suggests POIs that are more novel and with larger reward, i.e., they are estimated to be more relevant, compared to SKNN. Hence, it was conjectured that in an on-line scenario, i.e., when users really interact with the RS, they will like more the novel and rewarding recommendations generated by Q-BASE. It was also conjectured that the higher precision of SKNN is due to its bias to recommend popular items, i.e., items that have been often chosen by the observed users.

Hence, motivated by the above mentioned studies, we here attempt to address the following research questions:

- **RQ1** *If SKNN achieves high precision by being biased towards popular items, can Q-BASE be modified, by biasing the recommendations towards more popular items, to achieve similar precision as SKNN?*
- **RQ2** *Will on-line users like the precise recommendations of SKNN more than those generated by Q-BASE, which are more novel and yet relevant?*

In this study, we focus on the analysis of these research questions. We propose a novel and flexible RS, called Q-POP PUSH, that is derived from Q-BASE and generates recommendations that optimises simultaneously two criteria: the reward of the recommendations (as for Q-BASE) and their popularity. We test Q-POP PUSH in an off-line experiment by comparing its performance with Q-BASE and two Nearest Neighbour next-item RSs: SKNN and s-SKNN (Ludewig and Jannach 2018). Finally, we assess in a user study the user perception of the recommendations generated by the best performing (off-line) models.

Our research extends previous analyses that evaluated the popularity bias in RSs (Abdollahpouri et al. 2019, 2020; Park and Tuzhilin 2008; Jannach et al. 2015). But, while most of the research has focused on measuring such a bias and taming it, we are interested in understanding the dependency between precision and popularity bias, as it is investigated in Jannach et al. (2015), and in devising methods that allow us to balance the pros and cons of this bias by enabling the selection of the right amount of recommendations' popularity that better suits a specific application scenario. Hence, we are interested in understanding the positive effect of the popularity bias in tourism RSs.

In the off-line analysis, we replicate the evaluation procedure that was originally used in Massimo and Ricci (2018), while here the research focus is question **RQ1**. We measure the off-line performance of the IRL-based and nearest neighbour RSs in terms of reward (estimated relevance), precision and popularity. We note that popularity is normally considered as an indicator of (lack of) novelty: a

popular item is unlikely to be perceived by a user as a novel one. We investigate the effect of the proposed hybrid approach Q -POP PUSH and we inspect whether the popularity bias of Q -POP PUSH can help to generate recommendations similar to those computed by SKNN. To this end, we also compare the Jaccard similarity of the recommendations produced by Q -BASE and Q -POP PUSH with the recommendations produced by SKNN.

Then, in a user study, we address the research question **RQ2**. We designed an interactive on-line system to measure the user-perceived novelty and appreciation for the recommendations generated by Q -BASE, Q -POP PUSH and SKNN. In this on-line system, the user can enter a set of POIs, which has already visited in a city (Florence), and she will obtain suggestions for possible next POI visits in the same city.

The findings of the above-mentioned analyses (off-line and on-line) are here summarised. Firstly, the off-line evaluation results show that Q -BASE generates next-POI recommendations with higher reward and lower popularity, but also with lower precision, if compared to SKNN and s -SKNN. The experiment addresses the first research question by showing that Q -POP PUSH can actually achieve a precision very close to that of SKNN and s -SKNN at the cost of a significant increase of the popularity of the recommendations. It is also interesting to note that Q -POP PUSH can be tuned, via its parameter, to obtain different precision-popularity trade-offs.

Secondly, the user study results confirm that while Q -BASE suggests more novel items, actually SKNN and Q -POP PUSH suggest items that the users like more. We explain this result by observing that items that are novel to the user are also hard to evaluate. Hence, it is difficult for the user to formulate an explicit appreciation (“like”) for an item that is discovered the first time with the help of the RS, and it is illustrated only by a picture and a short description. But, the user study shows that Q -BASE can identify, better than SKNN and Q -POP PUSH, items that are both novel and liked by the users. This is an important result, since a valuable goal of an RS, especially in the tourism domain, is to identify novel experiences that are liked by the user.

Here we summarise the contribution of this study:

- We address the next-POI recommendation problem which is an important functionality for tourism RSs and not yet implemented and deployed by the on-line tourism industry.
- We describe two inverse reinforcement learning based (IRL-based) RSs that try to excel on different target criteria: precision, relevance and novelty.
- We devised off-line and on-line comparative studies of the IRL-based RSs with session-based RSs, applying them to the tourism domain.
- We show that the difference in performance between the IRL-based and the nearest neighbour-based RSs lies in the popularity bias of the latter.
- We investigate the effect of the popularity bias by showing that with an opportune definition of the model parameter the IRL-based RS Q -POP PUSH can achieve a precision similar to that obtained by the nearest neighbour-based RSs, and yet more reward.

- We show that the on-line user perception of the recommendations generated by the compared models is strongly influenced by the novelty of the recommendations.

The paper structure is as follows. In Sect. 2 the most relevant and related research works are presented. Then, in Sect. 3 we describe how IRL-based recommendations are generated and the novel hybrid model Q-POP PUSH is introduced. The experimental data used in this study is presented in Sect. 4. Section 5 presents the off-line evaluation of the proposed IRL-based models. The user study is the subject of Sect. 6. In particular, the on-line system developed for the user evaluation of recommendations is here presented. The outcomes of the user study are presented in Sect. 7. Finally, in Sect. 8, the limitations of the proposed approach are discussed and conclusions are stated.

2 Related works

POI recommendation has drawn considerable attention in RS research, but research results have not yet reached the level of development needed to be applied in tourism portals. In fact, major players of the on-line tourism market, such as, Booking.com or Tripadvisor.com, still offer a recommendation functionality that is not personalised: it is either based on the average opinion of the users or on experts' opinions or on the items' popularity.

In Torrijos et al. (2020) POI recommendations are generated by leveraging POI-visit trajectories collected by the Foursquare Location-Based Social Network (LBSNs) and by employing a user-based nearest neighbour algorithm. The authors propose to select neighbour trajectories by leveraging state of the art similarity measures, previously used in the field of trajectory data mining: Dynamic Time Warping and Hausdorff Distance. However, these similarity measures solely consider spatial-temporal properties of the trajectories and ignore the semantic of a POI-visit, which is instead considered in the RSs models that we propose (Q-BASE and Q-POP PUSH). Moreover, the recommendation generation process is agnostic of the sequential nature of next-POI selection.

Another POI recommendation method, which exploits POIs specific geographical information is presented in Wang et al. (2018). The proposed approach is also assessed by employing check-in data collected by LBSNs. The authors leverage the capacity of a POI to spread visitors to or attract them from other POIs. The authors also consider the physical distance among POIs. The proposed method uses the above mentioned data to compute the probability of a user to visit a POI. It is important to note that, differently from Q-BASE and Q-POP PUSH, the POI content information and the visit context is not used.

In Zhang et al. (2018) is presented a personalised RS for a specific type of POIs, namely, restaurants. The authors propose a clustering technique that identifies in TripAdvisor groups of related customers and groups of related restaurants. Restaurant recommendations for a target user are computed in two steps. Firstly, the customer group that is closer to the target user is identified. Then, the most related restaurants

are proposed. We note that this RS requires explicit user feedback to generate recommendations while in Q -BASE and Q -POP PUSH only behavioural data (implicit feedback), i.e., observations of POI-visit trajectories are considered.

It is also worth stressing that the previously described approaches do not consider the sequential nature of the tourists' choices while building an itinerary. This aspect has been instead considered in Palumbo et al. (2017), where a neural network-based next-POI RS is presented. The authors, also in this case, use check-in data collected by LBSNs. Besides, they consider users' demographic to cluster their check-ins. A Recurrent Neural Network is trained to identify common visit patterns for the (specific) clustered check-ins. Relevant next-POIs recommendations are then found by using the POI category that the model predicts. The usage of clustering in Palumbo et al. (2017) solves the new user problem, but in contrast to Q -BASE and Q -POP PUSH, it is not paired with a learning step where the user's behaviour model is derived.

In Huang and Gartner (2014) is presented a next-POI RS that, similarly to our approach, exploits observations of sequences of POI-visits made by users. But, differently from Q -BASE and Q -POP PUSH, the visit context and POI specific information are not leveraged in the generation of the recommendations. Next-POI recommendations are computed by identifying users similar to the target one. These are users that visited the last visited POI of the target user, and after that, also visited POIs not yet visited by the target user.

A basic and important difference between the state-of-the-art approaches discussed so far and that proposed in this article, is related to the fact that these approaches do not learn an explicit tourist behavioural model, i.e., a model that can predict and explain why the tourist is making their POI-visit choices. The previously mentioned approaches mine frequent patterns in the observed users' data without seeking for structural properties correlated to the user choices, i.e., they do not try to reveal the factors that steer users to take specific actions.

Inverse Reinforcement Learning (IRL) is a machine learning technique that can address the above mentioned task by finding the optimal policy of a Markov Decision Process (MDP) where the reward function is unknown (Ng and Russell 2000). In practice, with IRL one can compute the optimal decision making policy of a decision maker modelled by an MDP by just observing the decisions that she makes, i.e., without knowing or making any assumption on the reward obtained by the decision maker during the process. Moreover, IRL estimates the reward function, i.e., it explicitly provides a function that measures how much a choice, in our case a POI-visit decision, is estimated to be rewarding for the decision maker.

We mention here a couple of applications of IRL to problems that are similar to the one we have considered, i.e., the decision making process of a tourist. In Ziebart et al. (2008) the authors apply IRL to road navigation. They identify a choice distribution over decision sequences (i.e., driving decisions) that matches the reward estimated from the demonstrated behaviour. This technique is useful to model route preferences as well as to predict destinations based on partial trajectories. Instead, in Suzuki (2018) the authors apply IRL to model pedestrian behaviour from observed traces. The learnt behavioural model is used to generate synthetic trajectories at the city level and conduct simulations for urban planning. The IRL-based solution was

shown to better simulate users' movements, compared to a popular baseline used in the sector of mobility and transportation. We must observe that these works focus on learning a decision making model, but they do not apply it to the recommendation task, which is instead the focus of our work.

Regarding the RS evaluation methods that are used in our work, it is important to cite other studies that have touched some of the issues discussed in this paper. In Kouki et al. (2020) the authors note that off-line evaluation does not provide enough information to identify the best RS, for all the considered metrics. They also note that the precision metric scores high an algorithm only when it predicts the exact same item that the user choose. In a practical scenario, however, there are near-identical products which, although they are assigned different identifiers, they should be considered as equally-good recommendations. This observation is motivating our adoption of the IRL approach and the introduction of the reward metric, which is discussed in the rest of the article. This metric is scoring high the items that have the properties of the items, which are typically chosen by a user in a context, but they might not be the same that were actually selected. In Kouki et al. (2020) the authors also conduct a user study, but while in their case the recommendations were evaluated by experts, on behalf of the real users, we directly asked the subjects who received the recommendations to evaluate them. Moreover, we must observe that the application domain is very different, as they focused on the home-improvement domain. Hence, the results of our analysis are novel and provide an additional perspective on the topic of how to effectively evaluate RSs. Another interesting study that obtained some results, which we have derived as well, is described in Loepp et al. (2018). Here the authors investigate the effect of consuming the recommended items on the user evaluation of the recommendations. They show that it is not always possible to adequately measure user experience without allowing users to consume items; this is observed in the music domain. Participants rated system effectiveness, choice and overall satisfaction higher when they could listen to music tracks prior to filling in the questionnaire. We have observed a similar situation in the tourism domain, and this can justify the observed inferior number of likes produced by the algorithms that suggest more novel items.

Finally, its worth discussing the relationship between item novelty and recommendation satisfaction. In Knijnenburg et al. (2014), it has been shown that users' choice satisfaction for an RS is influenced by the users' knowledge about the recommendation domain. In that study the analysis was conducted on a different application domain, namely, the energy domain. But, the general principle can be applied to the tourism scenario as well: the knowledge of a destination influences the evaluation of a tourist for POI recommendations in that destination. Hence, a tourist needs not only suggestions of POIs that she may like, but also recommendations that she can recognise that she will like. In order to achieve that goal, the recommendations must match the user's level of knowledge of the domain and therefore must be recognisable as relevant suggestions by the user. In practice, this means that the user should have the possibility to estimate, based on her knowledge, that the offered recommendations match her needs and wants, hence, recommendations cannot be too much novel and diverse from those previously consumed. In Ekstrand et al. (2014) the authors report the

Fig. 1 Proposed approach for next-POI recommendation

outcome of a user study where they asked users to compare lists produced by three common collaborative filtering algorithms on the dimensions of novelty, diversity, accuracy, satisfaction, and degree of personalisation, and to select an RS that they would like to use in the future. They found that satisfaction is negatively dependent on novelty and positively dependent on diversity. This, further stresses the difficulty to generate recommendations that are novel and liked, which is one of the primary goals of our RS.

3 Recommendation techniques

Figure 1 shows the logical computational phases of the proposed Next-POI recommendation approach. These phases are described in the next sub sections.

3.1 User behaviour modelling

We model the user (tourist) choice-making behaviour with a Markov Decision Process (MDP) (Sutton and Barto 1998). A MDP is a tuple (S, A, T, r, γ) . S is the set of possible states; in our scenario, a state models the visit to a POI in a specific contextual situation. For instance, a tourist who visits Florence could be at Ponte Vecchio (POI) in an overcast, mild morning. So, a state is the combination of a physical location and a contextual situation. The specific contextual dimensions that we use are weather (sunny, rainy or windy time); day time (morning, afternoon or evening); and visit temperature (warm or cold). A is the set of actions a user can perform: the action space. In our scenario, an action represents the decisions to move to a next POI (next state). Consequently, POIs and actions are in a bijective relation. We assume that a user located at a specific POI and context can potentially reach any other POI in a possibly new context (e.g., the day time may change). T is the (finite) set of transition probabilities. A transition probability $T(s'|s, a)$ quantifies the chance to perform a transition from state s to s' when action a is taken. For instance, a user that visits the Uffizi Gallery in a cloudless morning (state s_1) and then wants to visit San Miniato al Monte (action a_1) in the afternoon, can arrive at the desired POI with either rainy weather (state s_2) or an overcast sky (state s_3). Transition probabilities are estimated by mining a data set of visit trajectories. The function $r : S \rightarrow \mathbb{R}$ models the reward a user obtains from visiting a state, i.e., visiting a POI in a particular context. This function must be learnt, i.e., the reward function is unknown in our application scenario. We assume that the reward the user obtains from a POI visit is unknown because the user is not supposed to reveal if that was a rewarding experience. Moreover, we assume that the reward obtained by the user is related to the relevance of the visit and therefore we believe that the estimated reward function reveals also what POI-visits (states) are relevant for the user. It is also worth noting that the learning approach that we use to learn the reward function, namely Inverse Reinforcement Learning, implicitly assumes that if the user visits a POI and not another nearby one, then this indicates that the first POI gives to the user a larger reward than the second. This is a very common assumption made in many RSs that exploit implicit feedback (Jannach et al. 2018).

Finally, $\gamma \in [0, 1]$ is a parameter measuring how much the rewards obtained from visits performed later in a visit trajectory are discounted with respect to the immediate ones: a reward received k visits after the current visit is worth only γ^{k-1} times what it would be worth if it were received immediately. The lower the value of γ the more myopic is the decision maker, i.e., she is just trying to optimise the immediate reward and less the rewards that can be obtained by the subsequent visits.

3.2 User behaviour learning

Given a MDP the objective is to find a policy $\pi^* : S \rightarrow A$ that maximises the cumulative reward that the user, i.e., the decision-maker, obtains by performing actions that follow that optimal policy π^* . The state-action value function $Q_\pi(s, a)$ expresses the value of taking the action a in state s under the policy π . It is the expected discounted cumulative reward obtained by taking action a in state s and then following the policy π . It is computed as $Q_\pi(s, a) = \mathbf{E}_\pi^{s,a}[\sum_{k=0}^{\infty} \gamma^k r(s_k)]$. The optimal policy π^* dictates to a user in state s to perform the action that maximises Q . In order to compute Q_{π^*} the previous formula is rewritten as: $Q_{\pi^*}(s, a) = \sum_{s'} T(s'|s, a)[r(s) + \gamma \max_{a'} Q_{\pi^*}(s', a')]$. When the reward function r is known, various reinforcement learning algorithms have been proposed to compute the optimal policy for a MDP (Sutton and Barto 1998).

With ζ_u we denote a sequence of actions performed by a user u , i.e., her POI-visit trajectory: ζ_u is a temporally ordered list of states (POI-visits). For instance, $\zeta_{u_1} = (s_{20}, s_3, s_{11})$ represents a trajectory of user u_1 , starting from state s_{20} , then moving to s_3 and ending at s_{11} . Z denotes the set of the available users' trajectories; they are used to estimate the probabilities $T(s'|s, a)$.

As we previously mentioned, in an RS the reward a user obtains by consuming an item is often not known because users tend not to provide explicit feedback on the consumed items (visited POIs) (Smith et al. 1996; Jannach et al. 2018). Moreover, the explicit feedback is often not faithfully describing the true user's preferences, as users are biased in how they remember their experiences (Do et al. 2008) and they do not report good and bad experiences with the same probability (Yang et al. 2018). Therefore, standard Reinforcement Learning techniques cannot be directly employed to compute the optimal decision making policy for a given MDP. Conversely, by having at disposal a good amount of POI-visit observations of a user, i.e., the user's POI-visit trajectories, a MDP (for each user) could be solved via Inverse Reinforcement Learning (IRL) (Ng and Russell 2000). IRL algorithms can learn simultaneously the reward function and its corresponding optimal policy. The optimal policy dictates actions close to the demonstrated behaviour, i.e., the observed user's POI-visit trajectory. Hence, in this context optimal is not referring to the actual best choice that a tourist can do, but it is optimal conditioned by the shown interests of the user. In this study, to learn the users' policy and reward function (behaviour), we have used Maximum Likelihood IRL (Babes et al. 2011). In fact, Maximum Likelihood estimation has been shown to be an effective optimisation criterion for solving IRL problems similar to those we address in this paper (Babes et al. 2011; MacGlashan et al. 2015).

3.3 Clustering trajectories

In general, it is hard to have at disposal a large user's history of travel-related choices, which is needed to learn the specific reward function of the user. The scarcity of users' specific preference and choice data is a common problem of e-commerce

Table 1 POI-visit trajectory example listing the visited POIs, their visit order (step) and the context of each POI visit: weather summary, temperature and part of the day

POI	Weather	Temperature	Day part	Step
Colonna dell'Abbondanza	Cloudy	Cold	Afternoon	1
Museo dell'Opera del Duomo	Cloudy	Cold	Afternoon	2
Cupola del Brunelleschi	Cloudy	Cold	Afternoon	3
Mercato Centrale	Cloudy	Cold	Afternoon	4
Piazzale degli Uffizi	Clear	Cold	Evening	5
Ponte Vecchio	Clear	Cold	Evening	6

Table 2 Document-like representation of the POI-visit shown in Table 1

POI-visit document-like representation	
(Step 1)	Cold, afternoon, cloudy, square, monument, foggini, century_15,
(Step 2)	Cold, afternoon, cloudy, museum, musei_storici, ghiberti, brunelleschi, century_15,
(Step 3)	Cold, afternoon, cloudy, church, brunelleschi, century_15,
(Step 4)	Cold, cloudy, afternoon, square, century_19,
(Step 5)	Cold, evening, clear, square,
(Step 6)	Cold, evening, clear, bridge, century_14

applications and RSs in particular; users may have not consistently interacted with the RS and, even if they have a large interaction history, they may not be eager to disclose it. In travel and tourism applications some travellers consider this information personal and private, especially because it can reveal their past position (Smith et al. 1996; Perentis et al. 2015; Poikela et al. 2014).

Hence, to learn a user's behavioural model from a relatively small set of observations of POI-visit actions, we cluster the full set of trajectories Z in a few smaller groups of trajectories belonging to users with similar interests and then we learn one behaviour model per each group.

We define the interests of a user, as they are shown in a visit trajectory, by utilising the terms that describe the visited POI (e.g., category) and the visit context (e.g., weather) associated with each POI-visit that is contained in that user's trajectory. These terms are also considered as Boolean features of the POI-visits; by representing the presence or absence of a term in the POI-visit description. We then use a document-like representation of each POI-visit trajectory where the interest of the user for a term (feature) is assumed to be proportional to its frequency in the trajectory's POIs. For instance, if "cathedral" appears very often in the document representation of a visit trajectory, and "bridge" only once, we infer that the user has a stronger interest for the first compared with the second. By building this representation for all the POI-visit trajectories in Z we obtain a collection of documents (corpora) that we then use to identify the different groups of interests of the users who performed those POI-visit trajectories. To achieve this objective we employ topic modelling (Blei 2012). In total, in the data set considered in this article, there are

Table 3 Most important terms according to the *tf-idf* weights for the document-like representation shown in Table 2

Term	Cloudy	Cold	Afternoon	Square	Museum
<i>tf-idf</i>	0.486	0.484	0.286	0.119	0.092

$|Z| = 1663$ POI-visit trajectories and $F = 137$ POI-visit Boolean features (listed in Sect. 4), which are also the terms appearing (or not) in a document-like representation of a trajectory.

Table 1 shows an example of POI-visit trajectory and its document-like representation is shown in Table 2.

A topic is a group of closely coupled and related POI terms, i.e., POI categories and visits conditions (weather) that are often found together in the documents (trajectories) of the collection. Each term in a topic is associated with a score that defines its importance in the topic, hence we can represent a topic with a list of terms ordered according to their (descending) term-to-topic relationship scores. For instance, the 5 most important terms, for two topics that we identified, are: *morning*, *cold*, *square*, *palace*, *15th century*, for topic A; and *hot*, *afternoon*, *16th century*, *church*, *palace*, for topic B. Topic A identifies trajectories of users that prefer to visit open areas (square) and palaces in the initial part of the day (morning), whereas topic B is representative of users interested in indoor activities (visiting churches, palaces) mostly during milder afternoon.

In addition, each POI-visit trajectory in the corpora is associated to a topic according to a trajectory-to-topic score that measures how strong is this link. By using these association scores we define clusters of similar trajectories, which are performed by users with similar interests (in context). We create a cluster for each topic and assign to the cluster the POI-visit trajectories that have a trajectory-to-topic association score that is larger than a threshold. The size of the threshold can be used to determine the size of the resulting clusters. We note that a POI-visit trajectory can be assigned to more than one cluster. This enables to leverage more samples when the behavioural model (reward function and optimal policy) of each cluster is learnt via Inverse Reinforcement Learning (Massimo and Ricci 2018).

More precisely, we consider the document-like representation of a POI-visit trajectory as a (row) vector of terms so that the combination of these vectors form the rows of a matrix. If the document-like POI-visit trajectory does not contain a specific term then the corresponding element of the matrix has value 0, otherwise, it contains the number of times the term appears in the document. Then, to better represent the importance of each term that appears in a document-like POI-visit trajectory we substitute the corresponding matrix element with the term frequency-inverse document frequency (*tf-idf*) (Rajaraman and Ullman 2011) of the POI-visit trajectory term. We finally denote with D the *tf-idf* POI-visit trajectories matrix, which has dimension $|Z| \times F$: one row for each trajectory and one column for each term. In Table 3 the five most important terms, according to the *tf-idf* weights of the POI-visit trajectory shown in Table 2, are shown.

Table 4 Row of the H matrix containing the trajectory-to-topic association scores of the same trajectory used in previous tables

Cluster A	Cluster B	Cluster C	Cluster D	Cluster A
0.012	0.019	0.165	0.005	0.001

Table 5 Top 10 terms in the five topics extracted from the trajectory data set and number of trajectories assigned to each topic (cluster)

#Term	Cluster A	Cluster B	Cluster C	Cluster D	Cluster E
1	Morning	Hot	Cloudy	Warm	Freezing
2	Cold	Afternoon	Cold	Cloudy	Cloudy
3	Square	Century 16	Church	Century 14	Afternoon
4	Palace	Palace	Square	Church	Century 14
5	Century 15	Church	Century 13	Square	Palace
6	Century 13	Square	Palace	Building	Building
7	Church	Century 19	Rain	Palace	Century 13
8	Night	Century 13	Museum	Ponte	Church
9	Dante	Museum	Brunelleschi	Century 13	Foggini
10	Century 10	Brunelleschi	Century 15	Century 19	Century 19
#Traj.	368	339	341	297	153

Topics are identified by using Non-negative Matrix Factorisation (NMF) (Lee and Seung 1999; Jia et al. 2017). Given the matrix D , NMF computes two matrices H (of size $|Z| \times K$) and W (of size $F \times K$), of a lower rank than D , such that $D \approx HW^T$. The matrix H identifies K hidden topics (columns) and their association to each visit trajectory (row). In our experiments, we identified the optimal K by using the stability analysis method described in Greene et al. (2014). That method is based on the measure of the agreement between term rankings (top- m terms of the K topics) generated over multiple runs of the NMF topic-modelling algorithm. The agreement is computed employing a top-weighted ranking measure (average Jaccard) that gives stronger influence to highly ranked terms. Details can be found in Greene et al. (2014). By applying this analysis and searching in the range $\{1, \dots, 30\}$, we found that the optimal number of topics for our data set is $K = 5$.

Concerning cluster formation, similarly to Jia et al. (2017), we associate a trajectory to any of the 5 topics that has trajectory-to-topic association score larger than a given threshold τ . For each topic the threshold is set as the median of the values in the columns of H . For instance, in order to illustrate the trajectory-to-topic association, we refer again to the POI-visit trajectory example shown in Table 1. Table 4 shows the specific scores that associate this trajectory to the 5 identified topics. This trajectory is finally included in cluster C since the value of that score is larger than the threshold.

Conversely, by using the matrix W we can identify the most relevant terms of a topic by selecting the rows with the largest values. In Table 5 we show the 5

generated topics, corresponding to the 5 generated trajectory clusters, in terms of the top-10 features of each topic and the number of included POI-visit trajectories.

3.4 Recommending next-POI visits

In this section we describe a new next-POI RS, which is called Q-POP PUSH, and extends the IRL-based Q-BASE model that was introduced in Massimo and Ricci (2018) (where it was called CBR). We first recall the definition of Q-BASE.

In Q-BASE the behaviour model of the cluster the user belongs to is used to suggest the optimal action this user should take after the last visited POI. The POI visit actions a suggested to user in state s are those with the largest $Q_{\pi^*}(s, a)$ values (Massimo and Ricci 2018). Here π^* is the optimal policy for the users in the cluster. Hence, if the tourist will make any of these choices and will continue to make successive POI visits by choosing the actions with the largest Q value, which are recommended by Q-BASE, then the obtained cumulative reward will be maximised. Q-BASE is therefore a recommendation strategy that not only tries to suggest the most satisfying immediate next POI visit, but also the visits that the tourist will be able to make after that immediate next. Moreover, since the reward is estimated on the base of the POI and context features, Q-BASE can even recommend novel POIs, not yet visited by tourists, provided that they have the features of the POIs visited by the tourists in the same cluster, and are visited in the contextual condition typically preferred by the tourist in the same cluster.

The original recommendation strategy introduced in this paper is called Q-POP PUSH. This generates recommendations that optimize two criteria: the cumulative reward of the next-POI visit recommendation, as for Q-BASE, and the popularity of the POI. Q-POP PUSH computes two scores and then combines them before selecting the POIs that have the largest combined scores. The first score measures how much reward can be obtained by the consumption of the POI and the second one measures how popular is the POI in the users' trajectories data set. Moreover, Q-POP PUSH can differently balance the two scores, hence, the recommendations can be made more or less popular, as needed.

Given a state s and a POI visit action a we consider the state-action value function $Q_{\pi^*}(s, a)$ and the $pop(a)$ function, which is the number of occurrences of the POI-visit corresponding to action a , in the trajectories data set Z . These two functions are normalised with min-max scaling to range in $[0, 1]$. Finally Q-POP PUSH scores a POI-visit action a for a user in state s as following:

$$\begin{aligned} QPP(s, a) &= \frac{1}{\alpha \frac{1}{pop(a)} + (1 - \alpha) \frac{1}{Q_{\pi^*}(s, a)}} \\ &= (1 + \beta^2) \frac{Q_{\pi^*}(s, a) \cdot pop(a)}{(Q_{\pi^*}(s, a) + pop(a) \cdot \beta^2)}. \end{aligned} \quad (1)$$

This is the harmonic mean of the two scores $Q_{\pi^*}(s, a)$ and $pop(a)$. The harmonic mean is widely used in information retrieval to combine two scores in a single one, e.g., the harmonic mean of precision and recall is F1 (Manning et al. 2008). In the

Table 6 Dataset statistics

Number of users	1163
Number of trajectories	1663
Number of POIs	779
Number of POI-visits	21,129
User-trajectory ratio	1.43
Avg. trajectory length	12.6

equation above $\beta^2 = \frac{1-\alpha}{\alpha}$ and $\alpha \in [0, 1]$, while $\beta \in [0, \infty[$. These two parameters weigh the relative importance of $Q_{\pi^*}(s, a)$ and $pop(a)$. With $\alpha > \frac{1}{2}$ ($\beta < 1$) the popularity of a POI-visit action has a higher importance, whereas with $\alpha < \frac{1}{2}$ ($\beta > 1$) the state-action value $Q_{\pi^*}(s, a)$ is weighed more. When $\beta = 1$ ($\alpha = \frac{1}{2}$) equal importance is given to the two scores. The POI-visit actions recommended to the user are those with the largest QPP scores.

4 Experimental data

In order to address the two research questions formulated in the introduction (**RQ1** and **RQ2**) we employed a data set consisting of POI-visit trajectories in the city of Florence (Italy).⁴ These trajectories have been extracted from a pre-existing dataset described in Muntean et al. (2015). To build the original dataset the authors first collected from the Flickr⁵ photo-sharing platform a set of photos taken in the city of Florence, together with their metadata (i.e., user id, timestamp, geographic coordinates). Then they built users' photo albums, i.e., manually grouped geo-referenced photos taken by the same photographer, who took more than one picture. Afterwards, each photo was matched to an existing POI by exploiting the geo-reference field in the POI Wikipedia pages of Florence attractions. Specifically, for each POI in Florence, they searched for photos that fall in the circular area (radius of 100 m) centred at the POI coordinates. Finally, for each user multiple POI-visit trajectories were formed. A POI-visit trajectory contains POI-visits that are separated by a time difference smaller than 8 hours.

Since our objective is to learn the sequential choice-making behavioural model of tourists visiting a physical space, we limit our analysis to a subset of the original data set, by considering only trajectories that contain at least 5 POI visits. In Table 6 we report some important statistics about the dataset we finally considered. We note that the trajectories/users ratio is 1.43. In practice, the majority of the user's in this dataset have just one visit trajectory. This data scarcity makes learning a user-specific user behaviour impossible, and it justifies the clustering approach followed in this article (Sect. 3.3).

⁴ The dataset and code used in this study can be obtained from the authors upon request.

⁵ <http://www.flickr.com>.

Fig. 2 Frequencies of the POI features in the trajectories' dataset

The identified POIs are mainly cultural attractions, and we sought for an appropriate set of features. The Boolean features here described corresponds exactly to the terms that were used for clustering the visit trajectories (see Sect. 3.3). POI content-related and contextual features define the state model and determine the learned user's behaviour: both the reward function and the optimal policy are functions of the state. By using the POI Wikipedia page of each POI we manually labelled them with the content features in Fig. 2.

In total we collected 137 content features that are divided in three groups: 13 *POI category* features; 18 *historical period* features; 106 *historical person* related to the POI (only one per POI) features. We note that the features were considered to be Boolean (present vs. absent), also to ease the computation of the inverse reinforcement learning algorithm.

In the selection of POI category features we balance the need to discriminate the items while being able also to tackle the problem of scarce POI-visit data. In other words, we considered features that are shared by multiple POIs. Clearly, the selected features depend on the peculiar characteristic of the historical city centre of Florence.

In Fig. 3 we show the 14 Boolean context features that we identified and used in the experiments. There are 6 features describing the *weather summary*, 4 features for the *temperature* and 4 features to model the *daytime* at POI-visit time. The identification of these features for each POI-visit has been performed by

Fig. 3 Context features' frequencies in the trajectories' dataset

leveraging the timestamp (the date) and the geographical coordinates of the original POI's photo and then querying a weather service.⁶

The MDP that we derived from the trajectories' data has $|S| = 2317$ states and $|A| = 779$ distinct visit actions. The feature vector used for representing the states ϕ has dimension 151 (137 content and 14 context Boolean features), and the total number of observed transitions is $|T| = 14,233$.

5 Off-line algorithm analysis

5.1 Baseline recommendation algorithms

We compare the performance of the IRL-based RSs, Q-BASE and Q-POP PUSH, with two nearest neighbour-based baselines that were previously proposed, session KNN SKNN (Jannach et al. 2017) and sequential SKNN (s-SKNN) (Ludewig and Jannach 2018), and a popularity-based baseline, POP.

SKNN generates next-item (visit action) recommendations by leveraging the user current POI-visit trajectory and searching for similar trajectories in the dataset. Firstly, SKNN finds N_{ζ} , which is the set of the N most similar trajectories to the current user trajectory ζ . The similarity $c(\zeta, \zeta_i)$ between the current trajectory ζ and one in the dataset ζ_i is computed as the cosine of the angle of the two boolean vector representations of the trajectories (one boolean value for each possible POI included in the trajectory). Then the score of a candidate next-POI visit action a is computed as follows:

$$score_{sknn}(a, \zeta) = \sum_{\zeta_n \in N_{\zeta}} c(\zeta, \zeta_n) 1_{\zeta_n}(a)$$

⁶ <https://darksky.net>.

where $1_{\zeta_n}(a)$ is the indicator function of the next-POI visit action a in the set of POIs contained in ζ_n : it is 1 if a is included in ζ_n and 0 otherwise. SKNN finally recommends the actions with the largest scores.

s-SKNN extends SKNN by employing a linear decay function $w_{\zeta_n}(\zeta)$ to weigh more in the prediction formula a neighbour trajectory w_{ζ_n} that contains the user's last observed visit actions in ζ . The neighbourhood of the current user POI-visit trajectory is obtained as in SKNN, while the computation of the score of a POI visit action is:

$$score_{s-sknn}(a, \zeta) = \sum_{\zeta_n \in N_\zeta} w_{\zeta_n}(\zeta) c(\zeta, \zeta_n) 1_{\zeta_n}(a).$$

In that scoring formula the weighting function $w_{\zeta_n}(\zeta)$ that is used to take into account the order of the POI-visit actions in the current POI-visit trajectory ζ . The weight computed by $w_{\zeta_n}(\zeta)$ is higher if a more recent POI-visit in ζ is also present in the neighbour POI-visit trajectory ζ_n . For example, let us assume that the user POI-visit trajectory ζ has length 7 and that b is the most recent action in ζ also contained in ζ_n and it is located at position 3 of ζ (counting from the beginning of the trajectory ζ). Then the weight defined by the decay function is $w_{\zeta_n}(\zeta) = 3/7$. Also, s-SKNN recommends the actions with the largest scores.

We found by 5-fold cross-validation that the optimal number of neighbours to be used in SKNN and s-SKNN is close to the full cardinality of the data set ($|N_\zeta| = 1200$). Finally, we observe that since in our data set we have essentially one trajectory per user, a pure NN algorithm will not differ essentially from SKNN.

Finally, we also consider a simple RS named POP: it suggests as next-POIs those visit actions that are most frequent in the POI-visits trajectories dataset. This is a non personalised RS that we include in our analysis only for better understanding the popularity bias of the considered RSs, that is, in comparison with the most biased one that could be used.

It is important to note that the baseline methods that we have introduced are not exploiting the knowledge contained in the content and context features that the IRL methods are actually using. Hence, as we have already mentioned, the differences in the performance of the compared method are influenced by both the learning approach and the used knowledge.

5.2 Evaluation procedure and metrics

In order to compare the considered RSs, i.e., the IRL-based ones (Q-POP, PUSH and Q-BASE) with the baselines (SKNN, s-SKNN and POP), we have split the trajectories dataset into training and test sets. The training set contains a random sample of 80% of the complete dataset of POI-visit trajectories. In particular, the train set contains 80% of the trajectories in a cluster for the IRL-based models, since these models are learnt separately for each cluster, and 80% of the full dataset for the nearest neighbour-based RSs and POP. The test set consists of the remaining 20% of the trajectories (either in the cluster or in the full data set). Each trajectory in the test set is further split temporally into an initial part (the

first 70% of the visited POIs) used for the generation of the recommendations and an observed part (the remaining 30% of the trajectory), which is used to compute the performance metrics. Here we refer to the initial part of the test trajectory used to generate the recommendations as ζ and we denote with Y_ζ the list of observed (next) POI-visit that are in the last part of the trajectory.

Let us denote with $R_{\zeta,s}$ a list of next-POI visit action recommendations for the user's POI-visit trajectory ζ in the state s , which is the last state in the POI-visit trajectory ζ . In order to evaluate the recommendation performance of the generated recommendations, $R_{\zeta,s}$, as usual, we consider Y_ζ , the observed behaviour for the test trajectory ζ . Let $a_o \in Y_\zeta$ be, in particular, the observed POI-visit action that the user has followed immediately after the last state s in the trajectory ζ . Moreover, with $\mathbb{I}_{Y_\zeta}(a)$ we denote the indicator function of the set Y_ζ : it is 1 if the action $a \in Y_\zeta$ and 0 otherwise.

We now define the four evaluation metrics that we have used to assess the recommendation performance of the proposed methods: reward, as defined in Massimo and Ricci (2018), popularity, precision and recommendations' similarity (to SKNN).

Reward: The goal of an RS is to satisfy the user by offering recommendations (POIs) that are relevant. Our IRL-based RSs estimate relevance with the reward function, hence it is interesting to measure the reward of the recommendations, and compare it with the reward of the next POI visit that the user has chosen, i.e., that are contained in the user's observed behaviour (Y_ζ). We compute the reward metric as following:

$$\text{reward}(R_{\zeta,s}, a_o) = \frac{\sum_{a \in R_{\zeta,s}} Q_{\pi^*}(s, a) - Q_{\pi^*}(s, a_o)}{|R_{\zeta,s}|}.$$

A positive score for this metric signals that the recommended POI-visits are more rewarding than the user action a_o , which is observed in state s , after having performed the initial trajectory ζ . Whereas a negative score indicates that the observed user action is more rewarding than the recommendations.

Popularity: An effective RS, especially in the tourism domain, should help users in identifying some novel items. The value of building sophisticated RSs just to recommend well known POIs is clearly limited. First of all, the user is likely to know them already. Secondly, even if sometime it is useful to pinpoint POIs that users may know, simple algorithmic solutions can be used for recalling users to visit the most popular attractions.

However, understanding that a POI is novel for the user is hard to be determined without querying the user. Therefore, in the off-line study we measure a proxy of novelty, i.e., popularity. We assume that a POI is more likely to be novel for a user if it is not popular. Let be $\text{pop}(a)$ the frequency of the POI-visit action a computed by considering the (training) data and let $\text{pop}_{\max}$ be the maximum value

of pop in the dataset. We then compute the popularity of a list of recommendations $R_{\zeta,s}$ as follows:

$$popularity(R_{\zeta,s}) = \frac{1}{|R_{\zeta,s}|} \sum_{a \in R_{\zeta,s}} \frac{pop(a)}{pop_{max}}.$$

A high popularity score (close to 1) indicates that the RS suggests POI-visits that are probably not novel. Whereas, a low popularity (close to 0) suggests that the recommended items may be unknown to the user.

Precision: Precision is largely adopted in the off-line evaluation of RSs, and it is defined as the fraction of the recommendations that are present in the user test set (Ludewig and Jannach 2018), i.e., that can be found in the observed user's behaviour. The precision of a list of next-item recommendations is then computed as following:

$$precision(R_{\zeta,s}) = \frac{1}{|R_{\zeta,s}|} \sum_{a \in R_{\zeta,s}} \mathbb{1}_{Y_u}(a).$$

Similarity to SKNN: An additional and useful evaluation metric is the similarity of the recommendations generated by the IRL methods with those suggested by the baseline model SKNN. SKNN was shown in previous studies to have high precision, but suffers from the popularity bias (Massimo and Ricci 2018). Hence, it is interesting to understand how much the IRL models deviate from SKNN to achieve their specific performance: low popularity and high reward.

Let $R_{\zeta,s}^A$ and $R_{\zeta,s}^{SKNN}$ be two recommendations lists of the same size that are generated for the same observed POI-visit trajectory ζ and current user state s , by the recommendation model A (e.g., one among the proposed models) and by SKNN, respectively. We compute the similarity of the two lists by using the Jaccard index:

$$sim_{KNN}(R_{\zeta,s}^A, R_{\zeta,s}^{SKNN}) = \frac{|R_{\zeta,s}^A \cap R_{\zeta,s}^{SKNN}|}{|R_{\zeta,s}^A \cup R_{\zeta,s}^{SKNN}|}.$$

5.3 Off-line study results

We now address the first research question **RQ1**, i.e., whether the better precision of SKNN with respect to Q-BASE, is determined by the popularity bias of the first and if Q-BASE can obtain the precision of SKNN with a modification that will bias it to recommend more popular items (Q-POP PUSH). Table 7 shows the quality of the Top-1, Top-5 and Top-10 recommendations of the considered RSs. Here, the effect of the α parameter on Q-POP PUSH is shown. We recall that when α is approaching 0, Q-POP PUSH behaves like Q-BASE, while for α approaching 1, Q-POP PUSH tends to recommend more popular items.

Table 7 Recommendation performance of the considered RSs

	Q-BASE	Q-POP PUSH				SKNN	s-SKNN	POP
		$\alpha = 0.009$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.8$			
Rew@1	0.073*	0.060	0.015	-0.002	-0.002	-0.007	-0.009	-0.310
Prec@1	0.043	0.070	0.105	0.099	0.096	0.109	0.109	0.000
Pop@1	0.397*	0.555	0.846	0.877	0.877	0.728	0.719	1.000
SimKNN@1	0.085	0.150	0.388	0.424	0.420	-	0.791	0.000
Rew@5	0.032*	0.020	-0.001	-0.009	-0.009	-0.010	-0.010	-0.015
Prec@5	0.045	0.062	0.063	0.060	0.060	0.068	0.063	0.050
Pop@5	0.319*	0.517	0.634	0.643	0.643	0.528	0.570	0.733
SimKNN@5	0.061	0.307	0.441	0.451	0.450	-	0.530	0.352
Rew@10	0.015*	0.003	-0.008	-0.011	-0.011	-0.012	-0.013	-0.014
Prec@10	0.036	0.054	0.058	0.058†	0.058†	0.052	0.055	0.050
Pop@10	0.259*	0.488	0.536	0.538	0.538	0.489	0.516	0.594
SimKNN@10	0.038	0.455	0.559	0.560	0.560	-	0.354	0.464

Best value for each metric is shown in boldface

*There is a significant difference ($p < 0.05$) between the best performing model and SKNN (two-tailed paired t-test). The test is run for the metrics Rew, Prec and Pop for 5 repeated train-test split

†There is a significant difference ($p < 0.1$) between the best performing model and SKNN

Clearly, SKNN and s-SKNN have the largest precision in all the three considered recommendation tasks (Top-1, Top-5 and Top-10), whereas Q-BASE suggests much less popular and higher reward items. The two nearest neighbour based RSs perform similarly, hence, in this dataset, the sequence-aware extension of SKNN seems not to offer an advantage. It is interesting to compare these two nearest neighbour based RSs with POP, i.e., a non personalised RS that simply recommends the most popular POIs. In the Top-1 recommendation task POP performs very poorly, with zero precision and very negative reward. But, interestingly, when the recommendation size grows (Top-5 and Top-10) POP produces recommendations that have a good overlap with SKNN (see the *SimKNN* metric) and it approaches the precision of SKNN. It is also interesting to note that the popularity metric of POP is equal to 1 in the Top-1 recommendation task and goes to 0.594 for the Top-10 task. This is due to the fact that less and less popular POIs can be included in longer recommendation lists. In conclusion, while the two nearest neighbour based RSs produce recommendations with large popularity, their behaviour, especially in the Top-1 and Top-5 tasks is quite different from a simple non personalised purely popularity based method, such as POP. But, when more recommendations are generated, as in the Top-10 task, the performance of the two approaches are very close.

Focusing now on Q-POP PUSH, we can complete our discussion of the first research question. In fact, when a popularity bias is added to Q-BASE, as it is done in Q-POP PUSH, the model generates recommendations that are more similar to SKNN, according to the *SimKNN* metric. By increasing α , the popularity of the recommendations produced by Q-POP PUSH increases and it becomes even larger than that of SKNN and s-SKNN. However, the best precision (in the

Fig. 4 Performance of Q-POP PUSH for different values of its β parameter—Top-5 recommendations

Top-1 task) of Q-POP PUSH, namely 0.105, which is essentially the precision of SKNN and s -SKNN (0.109), is not obtained with the largest popularity bias, i.e., $\alpha = 0.8$, but with $\alpha = 0.1$. With this setting Q-POP PUSH has still a positive reward (0.015), while SKNN and s -SKNN have negative rewards. This means that the popularity bias in Q-BASE could be beneficial, but in order to improve the system's precision, the right amount of this bias must be identified. We must also note that with larger values of α the reward of Q-POP PUSH is becoming smaller and smaller, and approaching that of the two nearest neighbour methods. This also shows that Q-POP PUSH can be tuned to balance two objectives: precision and reward.

Regarding Top-5 and Top-10 task, Q-POP PUSH shows (essentially) the same behaviour discussed above. Interestingly, Q-POP PUSH in the Top-10 task is even more precise than the SKNN RSs.

Figure 4 precisely shows how the popularity, precision and reward of Q-POP PUSH (top-5 task) depend on the parameter α .

For $\alpha \in (0, 0.13]$ popularity quickly increases. Then it still grows in the interval $(0.13, 0.46]$ reaching a plateau at the maximal popularity score for $\alpha > 0.46$ ($\beta \leq 1$). The second chart clearly shows that one can optimise precision with a

Fig. 5 Preference elicitation UI detail

careful selection of the α parameter. In particular, there is a steep increase of precision for $\alpha \in (0, 0.13]$. Then, for $\alpha > 0.13$ the precision of the recommendations decreases, with an exception for $\alpha = 0.34$ ($\beta = 1.39$) for which we have a second peak. Subsequently, for $\alpha > 0.71$ there is a further drop in precision. Finally, the last chart shows that the reward of the recommendations monotonically decreases for increasing values of α .

6 On-line user study

To answer the second research questions **RQ2** we designed an on-line user study aimed at measuring the user perceived novelty and appreciation of the recommendations. We recall that in **RQ2** we ask if on-line users will like the precise recommendations of SKNN more than those generated by Q-BASE, which are more novel and yet relevant. In order to answer this question we have executed an on-line study and have compared Q-BASE, the hybrid model Q-POP PUSH, with parameter $\alpha = 0.5$, to the SKNN baseline. The choice of the parameter $\alpha = 0.5$ is motivated by the desire to give equal importance to the popularity of POIs and the reward of the next-POI visit, rather than to optimize Q-POP PUSH, e.g., with respect to precision. In fact, in the previous section we saw that the largest precision and reward is obtained with a smaller value of $\alpha = 0.13$.

We have developed an on-line system to assess the quality of next-POI recommendations offered to a user that has already visited some POIs. The system initially asks the user to enter as many as possible previously visited POIs (in Florence). These are used to associate her with a cluster of similar users' trajectories and to generate an hypothetical itinerary of five POIs that the user must assume she has already visited. Then, the system generates next-POI recommendations with the three tested RSs, combines them in a unique list, and asks the user to evaluate them. The user does not know which algorithm recommends each single displayed recommendation. The training data of the on-line system is the same previously used in the off-line study, i.e., the RSs are trained on five clusters of trajectories obtained from the original 1663 POI-visit trajectories over 532 POIs.

Fig. 6 Evaluation UI. From top to bottom: itinerary detail; info box; recommendations and item details

6.1 On-line system

The user-system interaction consists of five steps: (1) landing phase; (2) introduction to the experiment and start-up questions; (3) preference elicitation phase; (4) recommendations generation; and (5) evaluation of the recommendations.

Once the user accesses the on-line system (landing phase) she can select the preferred language (Italian or English). Afterwards, if the user accepts to participate in the experiment, the system asks whether the user has already been in Florence. If she replies “no” the procedure ends, because we believe that in this case the user cannot properly evaluate next-POI recommendations that extend a previously initiated itinerary. If the user has instead already visited Florence, the user is considered to have some experience of the city, and she can declare which POIs has previously visited. This is done in the next interaction phase. In Fig. 5 we show a detail of the user interface designed to support this phase. The selection of POIs can be performed in a mixed modality: the user can either seek for POIs utilizing a search-bar with auto-completion or by interacting with a selection pane that contains the 50 most popular POIs in the POI-visit trajectories dataset. To help the user to recognise or remember a visited POI is offered the possibility to tap/click on an entry of the selection pane and visualise a media card that provides information about the POI in the form of a picture together with a textual description. POI specific data has been extracted from Wikipedia. Each POI the user selects (visited) is added to an editable list that is considered to be the profile of the user. We note that the POIs added to the user profile are describing the user past behaviour, not the items liked in the past, exactly as in the trajectories data that is used to train the RS. In other words we tried

to match in this experimental design a feature of the proposed approach, i.e., leveraging only implicit feedback data.

Once the user has indicated the previously visited POIs, the system associates the user to one of the clusters of users' trajectories showing similar interests and generates a short itinerary composed of at most 5 POIs, among those in the user profile. An example of such an itinerary is shown on the top part of Fig. 6. The user should imagine having followed the composed itinerary and, at that point, to request next-POI recommendations. The system proposed next-POI recommendations are meant to complete the current visit itinerary of the user.

We note that we generate a fictitious itinerary because we do not want to ask the user to remember exactly any previous visit itinerary. However, by including in the itinerary POIs that are in the user profile, we also tried to generate a trajectory that the user is likely to have followed. By showing a hypothetical itinerary to the user we want to reinforce in the user's mindset the specific setting of the supported recommendation task: next-POI recommendation. Besides, to simplify the user's assessment of the recommendations, the context variables (time and weather) were not shown and not used in the recommendation generation process.

When finally the user evaluates the received recommendations she is presented with the GUI shown in Fig. 6. At the top of the page is shown the hypothetical (5-POIs) itinerary generated by the system and that the user should assume has followed so far. Finally, the participant is informed that in the bottom-left box there are next-POI recommendations that she could visit now. The user evaluates each POI by marking it with one or more of the following labels: "I already visited it" (eye icon), "I like it" for a next visit (thumb up icon) and "I didn't know it" (exclamation mark icon).

6.2 Recommendation list generation

In this section, we give additional details on the recommendation generation process. We recall that in our proposed approach, to generate recommendations with Q -BASE and Q -POP PUSH, a user-study participant must be associated with one of the 5 existing POI-visit trajectories clusters.

User-cluster pairing: To assign a user study-participant to one of the 5 existing clusters the system leverages the user profile. In fact, a k -nearest neighbour classifier (Hastie et al. 2001) is built by using as training data the tf-idf vectors of the POI-visit trajectories in the 5 clusters. This classifier takes as input the tf-idf vector representation of the user-study participant's profile and outputs the cluster with the highest probability to contain the user-study profile.

The optimal number of neighbours of the classifier has been identified by 10-fold cross validation grid search, in the range of $k \in \{1, \dots, 20\}$. The data was the split by allocating 80% of the dataset for training the classifier and the remaining 20% was used as the test set. The classifier's best configuration showed an accuracy of 67%. We note that the low performance of the classifier may have penalised Q -BASE and

Q-POP PUSH in the on-line study; in fact, the participant's user profile may have not been associated to the best cluster.

Recommendations: The next POI-visit recommendations are generated by leveraging Q-BASE, Q-POP PUSH and SKNN. In particular, the three RSs take as input the POI itinerary that is shown to the user and is assumed to be followed so far, and generate 3 recommendations each. Then, from the recommendations generated by the three algorithms, the POIs already present in the user profile are discarded. In fact, we do not want to suggest as next-POI any item that the user has already visited. Despite this, it is important to highlight that the user may still find in the recommendations some POIs that she has already visited, because she may have not indicated in the first phase of the user-system interaction all the POIs that she visited in the past (Smith et al. 1996; Perentis et al. 2015).

The top-3 recommendations generated by each RS are combined in order to produce the shown recommendation list. We have decided to use this evaluation approach, instead of implementing a between-group experimental design, where each user is testing a different RS, because it enables to acquire more evaluation data, without incurring into the limitations of a within-group experiment where a user evaluates, one after the other, the competing RSs. Moreover, it is worth noting that the adopted approach is very common in search engine evaluation (Joachims 2002).

To avoid biases in the recommendation evaluation phase, we do not reveal to the user which recommendation algorithm has produced which POI recommendation. To combine the three recommendation lists produced by the RSs, first, for each of the user-study participant, we generate a random order that we follow to pick items from the three lists of top-3 suggestions. Then, by following that order we aggregate the ranked lists by picking up, in turn, the items from the top to the bottom of the sorted lists. For instance, if the aggregation order is Q-POP PUSH SKNN and Q-BASE, then, the aggregated list of recommendations that is shown to the user, contains in the first position the top recommendation of Q-POP PUSH, then the top item suggested by SKNN and then that suggested by Q-BASE. The same procedure is applied for the remaining positions. In case a POI is suggested by more than one algorithm, the item is shown only once in the final list. Therefore, the list of next-POI visit recommendations shown to the user study participant contains at most 9 entries, if all the algorithms generate different POI-visit recommendations, and at least 3 items, if all the algorithms generate all the same POI-visit recommendations.

7 Results of the on-line user study

The on-line user study participants were recruited via social media and mailing lists. Over 300 subjects took part in the study; 202 of them declared to have previously visited Florence, hence only their data was finally used. We did not ask to the subjects any information (e.g., sex, age, citizenship, etc.); the participation to the experiment was totally anonymous. We then excluded unreliable replies, i.e., those surveys completed in less than 2 min, which was estimated as the minimum

Table 8 Example of a recommendation list and the collected user's evaluations (Source: authors)

Recommendations			Models			User feedback		
Nr.	POI name	POI pop.	Q-BASE	Q-POP P.	SKNN	Visited	Liked	Novel
1	Baptistry of San Giovanni	0.53		✓	✓	✓	✓	
2	Spedale degli Innocenti	0.05	✓				✓	✓
3	Basilica of Santa Croce	0.47		✓		✓	✓	
4	Torre dei Pulci	0.60			✓			✓
5	Statue of Ferdinando I de 'Medici	0.08	✓				✓	✓
6	Door of the Mandorla	0.55			✓	✓		
7	Brunelleschi's dome	0.43	✓			✓	✓	
8	Santa Croce's Square	0.34		✓		✓

Fig. 7 Probability that a user study participant evaluates as “liked”, “visited” and “novel” an item ranked at a specific position (Source: authors)

time to carefully complete the experiment. Finally, 158 subjects' data was kept. These users examined in total 1119 next-POI visit recommendations, computed by the three RSs SKNN, Q-POP PUSH and Q-BASE. A subject was exposed on average to 7.1 recommendations.

To give an idea of the recommendations offered to a user and her corresponding evaluation we illustrate an example of recommendation list in Table 8. The suggested POIs are listed on the rows of the table in the same order they were shown in the user interface at evaluation time. The table columns are: the POI name; the POI popularity; the model that produced the recommendation; and the user study participant's feedback, among “visited”, “liked” and “novel”. The next-POI recommendations shown in this example have been offered to a user that at recommendation time was supposed to have completed the following 5 POIs itinerary: Giotto's Bell Tower, Old Bridge, Cathedral of Santa Maria del Fiore, Piazza della Signoria, Fountain of Neptune.

The order used to aggregate the recommendations of the three RSs is: SKNN, Q-POP PUSH and Q-BASE. The first (top-ranked) item in the list is suggested by both Q-POP PUSH and SKNN. That item is marked as already visited and

Table 9 Probability to evaluate a recommendation of an algorithm as visited, novel and liked

	Q-BASE	Q-POP	PUSH	SKNN
Visited	0.165	0.245		0.238
Novel	0.517	0.376		0.371
Liked	0.361	0.464		0.466
Liked and novel	0.091	0.076		0.082

also liked by the user study participant. The second item instead is recommended by Q-BASE and is assessed as novel and liked by the user. We can observe that, for this specific subject, all the next POIs suggested by Q-POP PUSH and SKNN have rather large popularity, and almost all of them are actually marked as “visited”. For instance, Torre dei Pulci has popularity 60%. The recommendations generated by the Q-BASE recommendation algorithm are much less popular, e.g., Spedale degli Innocenti has popularity 5%. Moreover, we observe that this user liked more the items suggested by Q-POP PUSH and SKNN. Interestingly, among the POIs that the user marked as “novel” she liked only the least popular, which is suggested by Q-BASE.

Figure 7 shows how the assessment of the user study participants varies at the various ranking positions of the recommendations. In particular, we compute the probability that a user marks the POI shown at the i -th position of the recommendation list with the three types of feedback: “liked”, “visited” and “novel”. The probability that a user marks an item as “liked” is around 40% at all the first six positions of the next-POI recommendation list. Then, this probability drops. Instead, the probability to mark an item as visited decreases more uniformly as the rank position increases. Finally, the probability that a user marks an item as novel mirrors the “visited” feedback, so, while the probability to be “visited” decreases with the ranking, the probability to be “novel” increases.

In Table 9 we show the main results of our analysis. We recall that we are interested in addressing the research question **RQ2**, namely, to discover if the users, in a real next-POI selection task, will like the precise and popular recommendation more than those generated by Q-BASE, which are more novel and with higher estimated relevance (reward).

Here the probabilities that a user marks a next-POI recommendation as “visited”, “novel”, “liked” or both “liked” and “novel” are shown. These probabilities are estimated by dividing the total number of next-POI recommendations assessed as, “visited”, “liked”, “novel” and both “liked and novel”, for each algorithm, by the total number of recommendations shown and produced by the algorithm. We recall that a user marked as “liked” a recommendation that she judged as a good candidate to perform next (given the current itinerary followed by the user). Therefore, here a “like” is not a generic appreciation of the item, but should also take into account the context of the hypothesised itinerary of the user study participant, and the “like” feedback is likely to be provided by considering what items the user has already visited. By looking at these results we note that the POIs recommended by SKNN and Q-POP PUSH have the highest probability (24%) of being already visited, and the

Table 10 Probability that a user study participant likes a recommended POI-visit given that she visited, knew or is unaware of it

	Q-BASE	Q-POP PUSH	SKNN
P(Liked novel)	0.176	0.202	0.222
P(Liked visited)	0.256	0.310	0.283
P(Liked known and not visited)	0.717	0.810	0.806

lowest probability to be marked as novel. Q-BASE instead has the lowest probability to recommend items that have been already visited (16%) and the highest probability that a user marks the recommended items as novel (52%). Hence, these results are consistent with those obtained in the off-line study (see Sect. 5) where Q-BASE was shown to recommend less popular items.

Considering the user “likes” for the recommendations, our conjecture was that an RS with high relevance, estimated off-line by the reward metric, should also collect a large number of likes in the on-line user study. But, by looking at the results reported in Table 9 we note that the outcome is not confirming that conjecture. In fact, Q-BASE, which has the largest off-line-measured reward, suggests POI-visits that on-line users liked with the lowest probability (36%). Conversely, Q-POP PUSH and SKNN tend to generate next POI-visit recommendations that are more liked by the user (46%). Therefore, one could answer the research question **RQ2** by stating that users like more the precise next-POI recommendations of SKNN and Q-POP PUSH than the more novel and yet relevant ones that are offered by Q-BASE. Our explanation of this is related to the difference between a relevant and “like” recommendation. The first is estimated by observing real users’ behaviour, while the second is measured by observing users’ reactions to POI descriptions. Hence, a “like” is measuring the expectation of the user for an item (expected utility), while the reward is estimating the experienced utility: a picture of a visited POI is surely a signal of rewarding experience made with an item. Moreover, users are not able to perfectly estimate the satisfaction that could be obtained by visiting a recommended POI unless the user is in some way familiar with that POI. Hence, novel POIs will tend to receive lower likes independently from their actual relevance.

We show in Table 9 also the probability of an item to be judged “Liked & Novel”; this is the probability that a user likes a novel POI that the RS presents for the first time to the user. Q-BASE, which showed in the off-line analysis the highest reward and lower popularity, in the on-line user study is suggesting with the highest probability (0.09%) next POI-visits that are both novel and liked. This is a valuable property of Q-BASE.

We used a two-proportion z-test with a significance level of 0.05 to check if the RSs are equally perceived by the user study participants (null hypothesis). In particular, we test if the RSs perform equally in producing liked, novel and visited next POI suggestions. We have found that for the RSs SKNN and Q-POP PUSH we cannot reject the null hypothesis, hence we accept that they are perceived having the same performance ($p > 0.93$ for all tests). Conversely, we reject the null hypothesis when comparing Q-BASE and SKNN (all $p < 0.04$) and Q-BASE and Q-POP

PUSH (all $p < 0.03$), for all of the three types of feedback. Hence, users are not perceiving Q-BASE as equally performing as SKNN and Q-POP PUSH.

In Table 10 we show the probability that a user will like a recommendation given that: she knows the item but has not yet visited it (“Known & Not Visited”); she visited it (“Visited”); or the item is “Novel” for her. We first note that the novel POI-visit suggestions generated by the recommendation algorithms SKNN and Q-POP PUSH are liked more (20 and 22% respectively) than those produced by Q-BASE (17%). In fact, often Q-BASE suggests items that are both novel and, even if estimated to be relevant, are rather specific, hence harder to evaluate. For instance, Q-BASE suggests Porta della Mandorla which is a door of the Duomo. This POI can be perceived as a very specific item, many will not know that POI, and surely much less attractive than the Duomo itself. In fact, few participants, after the execution of the experiment, left a note declaring that it is difficult to like something that is unknown.

This is clearly shown by the fact that, for all the three RSs, the probability that a user likes a recommended next-POI that she *visited* tends to be much larger than the probability to like a *novel* one. More specifically, the probability of liking a POI given that it was visited is: 31% for Q-POP PUSH ; 28% for SKNN; and 26% for Q-BASE. We believe that the performance difference is again due to the fact that both SKNN and Q-POP PUSH tend to recommend known (not novel) POIs, i.e., which are easier to judge because users may have already heard them, whereas Q-BASE recommends relevant but niche items.

Finally, we discuss the probability that a user study participant will like an item that she *knows* but has not yet visited. In Table 10 we find a similar pattern as that observed before. Q-POP PUSH and SKNN suggest items that will be liked with a higher probability, respectively 81% and 80%, than Q-BASE (71%). It is interesting to note that these probabilities are very large. This clearly shows that knowing a POI is an important condition to like an item. In this case, the user can more reliably estimate the potential utility of visiting the POI, compared with an item for which the user has not information.

8 Discussion and future work

8.1 Summary of the obtained results

We have analysed the performance of new recommendation techniques, based on Inverse Reinforcement Learning (IRL), especially designed to solve the next-POI recommendation problem. At first, we have analysed the impact of the popularity bias on system performance (Abdollahpouri et al. 2019, 2020; Park and Tuzhilin 2008; Jannach et al. 2015; Massimo and Ricci 2018). We have formulated a specific research question, namely: “If SKNN achieves high precision by being biased towards popular items, can Q-BASE be modified, by biasing the recommendations towards more popular items, to achieve similar precision as SKNN?”. We have then developed a new hybrid IRL-based RS, Q-POP PUSH, that combines the Q-BASE POI scoring with a second score, which is proportional to the POI popularity. This

hybrid model tends to recommend more popular items than Q-BASE. We have shown, with an off-line experiment, that Q-POP PUSH can actually obtain better precision than Q-BASE, and reach the same precision of SKNN. Moreover, we have shown that Q-POP PUSH can actually be tuned to balance contrasting goals: precision, popularity and relevance of the recommended items. Hence Q-POP PUSH can be effectively used in an operational scenario.

Then, we have formulated a second research question, namely: “Will on-line users like the precise recommendations of SKNN more than those generated by Q-BASE, which are more novel and yet relevant?”. In order to reply to this question we have designed and conducted a user study that simulates the planning of a visit to the city of Florence (Italy). We show that Q-BASE generates next POI-visit suggestions that the users perceive as novel, but, also for this reason, not as liked as the competing RSs: SKNN and Q-POP PUSH. However, we show that Q-BASE generates more recommendations that are both liked and novel. Hence, Q-BASE may better accomplish an objective of tourism RSs, which is, recommending venues (POIs) that are novel and yet relevant for a user.

We also identified a reason why Q-BASE does not produce the recommendations that are liked the most. Q-BASE is designed to produce relevant recommendations by optimising the reward obtained by a user when visiting a POI (experienced utility), while the on-line study measured the perception of the user at the choice time (expected utility). Hence, when the POI is not known by the users, it is hard for them to assess the POI value and express a “like”. As other researchers have already noted in other application domains (Loepp et al. 2018), for a user, estimating how good (or bad) can be a future visit to a POI, which is not yet known, is a cognitively difficult task. This result has been supported by the analysis of the users’ reaction to POIs that were known, compared to those that were not known, before the system showed them.

8.2 Limitations and future works

The research described here has a number of limitations that it is important to discuss. First of all, we must stress that the technology and the evaluation conducted here relate to a system prototype that is surely not yet mature enough to be deployed in a operational system. Hence, our analysis focused on addressing general research questions on the performance of a class of technologies, such as nearest neighbor methods and inverse reinforcement learning approaches, by means of the analysis of the considered technology *in the lab*, i.e., not in a real operational environment.

We must then immediately observe that even in this restricted context, nowadays there are other recommendation technologies that can be applied to this task. In particular it will be important to consider in a future analysis especially Deep Neural Network models that have shown excellent performance in many domains. It will be important to understand if DNN models suffer from the same popularity bias and also if these models can, better than KNN, identify less popular items that are yet relevant for the user.

A second important limitation of our work is related to the mismatch between the optimisation procedure, which determines the optimal policy and the estimated reward function of Q -BASE, and the actual recommendation task, i.e., next-POI recommendation. In fact, Q -BASE, given a tourist in a state, is suggesting the POI visit actions a with the largest $Q_{\pi^*}(s, a)$ values (Massimo and Ricci 2018). Here π^* is the optimal policy for the users in the cluster. Hence, if the tourist will make this choice and will continue to make successive POI visits by choosing the actions with the largest Q values, which are recommended by Q -BASE, then the obtained cumulative reward will be maximised. Q -BASE is therefore a recommendation strategy that not only tries to suggest the most satisfying immediate next POI visit, but also the visits that the tourist will be able to make after that immediate next. We have not evaluated the quality of a succession of POIs that Q -BASE can actually recommend and which is it optimised for. This is an important assessment that should be completed; it may be the case that Q -BASE is not recommending the best next POI because is also considering the successive choices that the user can make in order to achieve a high cumulative reward in a complete itinerary.

Another important limitation of our analysis resides on the trajectory data. We derived this information from pictures, taken by tourists, and uploaded in a social network. The considered data set is not very large, it contains less than 2000 trajectories, and is focused on a particular city, and on a particular type of POIs (cultural attraction). It is worth noting that social network users do not represent the full spectrum of tourists, and the core problem of acquiring unbiased and representative behavioural data remains. Moreover, while we cluster the trajectories to determine distinguished groups of tourists we do not explicitly try to understand who these tourists are, i.e., for instance if they are local citizens or visitors. In fact, it has been recently shown that different groups (tourists and locals) can be better served by completely different RSs and that tourist behaviour data in another city can be used to improve the performance of the RS in the target city (Sanchez and Bellogín 2021).

Related to the previous topic, we observe that the state model that we have adopted and the POIs that we have considered may strongly influence the outcome of the evaluation. A more detailed description of the POIs, i.e., by using a larger number of features, and consequently a larger state space, should be evaluated. From one side it could help to estimate a more precise reward function, but at the same time one must cope with the growth of the state space and the increased complexity to learn the optimal policy and reward function. In fact, to limit the time and memory complexity of the model learning algorithm it is important to consider only a small set of relevant features. We note that the larger is the set of features which are used to define a POI visit, the harder it is for the learning algorithm to converge to a solution in a short time. Besides, the number of context features impacts directly on the memory complexity of the algorithm. In fact, the state space size of the MDP increases proportionally to the number of considered context features. In addition to that, the selection of the POIs is here determined by matching the geo-referenced pictures taken by tourists to POIs listed in Wikipedia. We may have erroneously matched pictures to POIs, i.e., we may have considered POIs that were not actually visited by the tourist that took the picture. This can explain the presence of rather

niche POIs that we have identified and considered in our study; these POIs may have a (Wikipedia) position close to that of the picture, but the picture may show another POI.

Finally, it is worth stressing here that the ultimate objective of this study is the development of an effective next-POI RS that employs a generalised tourist behaviour model learned from implicit data, i.e., observations of POI-visit action sequences. While in this work we have shown the benefits and limitations of such recommendation technology through an off-line analysis, and an on-line study based on an RS designed for these experiments, we have also understood the importance to conduct further user studies involving tourists while visiting a destination: this is necessary to test if the users will like the experience of the POI and not only the prospect to visit them.

Author contributions The paper presents the authors' original contribution.

Funding Open access funding provided by Libera Università di Bolzano within the CRUI-CARE Agreement. Not applicable.

Availability of data and material References to the publicly available data are given in the paper.

Code availability References to code availability are given in the paper.

Declarations

Conflict of interest There are no conflicts of interest or competing interests.

Consent to participate Users involved the live user study were explicitly informed about the study nature and asked about the consent for participation.

Consent for publication Users involved the live user study were informed about the fact that the collected data were going to be published in an aggregated form.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abdollahpouri H, Burke R, Mobasher B (2019) Managing popularity bias in recommender systems with personalized re-ranking. In: Barták R, Brawner KW (eds) Proceedings of the thirty-second international Florida artificial intelligence research society conference, Sarasota, Florida, USA, May 19–22 2019. AAAI Press, Palo Alto, pp 413–418

- Abdollahpouri H, Adomavicius G, Burke R, Guy I, Jannach D, Kamishima T, Krasnodebski J, Pizzato LA (2020) Multistakeholder recommendation: survey and research directions. *User Modeling and User-Adapted Interaction* 30(1):127–158
- Adomavicius G, Tuzhilin A (2015) Context-aware recommender systems. In: Ricci F, Rokach L, Shapira B (eds) *Recommender systems handbook*. Springer, Berlin, pp 191–226
- Babes M, Marivate VN, Subramanian K, Littman ML (2011) Apprenticeship learning about multiple intentions. In: Getoor L, Scheffer T (eds) *Proceedings of the 28th international conference on machine learning, ICML 2011, Bellevue, Washington, USA, June 28–July 2, 2011*. Omnipress, Madison, pp 897–904
- Blei DM (2012) Probabilistic topic models. *Communications of the ACM* 55(4):77–84
- Cheng C, Yang H, Lyu MR, King I (2013) Where you like to go next: successive point-of-interest recommendation. In: Rossi F (ed) *IJCAI 2013, Proceedings of the 23rd international joint conference on artificial intelligence, Beijing, China, August 3–9, 2013. IJCAI/AAAI, Palo Alto*, pp 2605–2611
- Do A, Rupert A, Wolford G (2008) Evaluations of pleasurable experiences: the peak-end rule. *Psychonomic Bulletin & Review* 15:96–98
- Ekstrand MD, Harper FM, Willemssen MC, Konstan JA (2014) User perception of differences in recommender algorithms. In: Kobsa A, Zhou MX, Ester M, Koren Y (eds) *Eighth ACM conference on recommender systems, RecSys '14, Foster City, Silicon Valley, CA, USA, October 06–10, 2014*. ACM, New York, pp 161–168
- Greene D, O'Callaghan D, Cunningham P (2014) How many topics? Stability analysis for topic models. In: Calders T, Esposito F, Hüllermeier E, Meo R (eds) *European conference on machine learning and knowledge discovery in databases, ECML PKDD 2014, Nancy, France, September 15–19, 2014. Proceedings, Part I. Lecture notes in computer science, vol 8724*. Springer, Berlin, pp 498–513
- Hastie T, Tibshirani R, Friedman J (2001) *The elements of statistical learning*. Springer series in statistics. Springer New York Inc., New York
- He J, Li X, Liao L, Song D, Cheung WK (2016) Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In: Dale Schuurmans and Michael P. Wellman (eds) *Proceedings of the thirtieth AAAI conference on artificial intelligence, February 12–17, 2016, Phoenix, Arizona, USA. AAAI Press, Palo Alto*, pp 137–143
- Huang H, Gartner G (2014) Using trajectories for collaborative filtering-based POI recommendation. *International Journal Data Mining Modeling Management* 6(4):333–346
- Jannach D, Lerche L, Kamehkhosh I, Jugovac M (2015) What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction* 25(5):427–491
- Jannach D, Kamehkhosh I, Lerche L (2017) Leveraging multi-dimensional user models for personalized next-track music recommendation. In: Seffah A, Penzenstadler B, Alves C, Peng X (eds) *Proceedings of the symposium on applied computing, SAC 2017, Marrakech, Morocco, April 3–7, 2017*. ACM, New York, pp 1635–1642
- Jannach D, Lerche L, Zanker M (2018) *Recommending based on implicit feedback*. Springer International Publishing, Cham, pp 510–569
- Jia X, Sun F, Li H, Cao Y, Zhang X (2017) Image multi-label annotation based on supervised nonnegative matrix factorization with new matching measurement. *Neurocomputing* 219:518–525
- Joachims T (2002) Optimizing search engines using clickthrough data. In: *Eighth ACM SIGKDD international conference on knowledge discovery and data mining, SIGKDD 2002, July 23–26, 2002, Edmonton, Alberta, Canada*. ACM, New York, pp 133–142
- Knijnenburg BP, Willemssen MC, Broeders R (2014) Smart sustainability through system satisfaction: tailored preference elicitation for energy-saving recommenders. In: *20th Americas conference on information systems, AMCIS 2014, Savannah, Georgia, USA, August 7–9, 2014. AIS/ICIS*, pp 1–15
- Kouki P, Fountalis I, Vasiloglou N, Cui X, Liberty E, Al Jadda K (2020) From the lab to production: a case study of session-based recommendations in the home-improvement domain. In: Santos RLT, Marinho LB, Daly EM, Chen L, Falk K, Koenigstein N, de Moura ES (eds) *Fourteenth ACM conference on recommender systems, RecSys 2020, Virtual event, Brazil, September 22–26, 2020*. ACM, New York, pp 140–149
- Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788–791
- Loepp B, Donkers T, Kleemann T, Ziegler J (2018) Impact of item consumption on assessment of recommendations in user studies. In: Pera S, Ekstrand MD, Amatriain X, O'Donovan J (eds) *12th ACM*

- conference on recommender systems, RecSys 2018, Vancouver, BC, Canada, October 2–7, 2018. ACM, New York, pp 49–53
- Ludewig M, Jannach D (2018) Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction* 28(4–5):331–390
- MacGlashan J, Babes-Vroman M, desJardins M, Littman ML, Muresan S, Squire S, Tellex S, Arumugam D, Yang L (2015) Grounding English commands to reward functions. In: Kavraki LE, Hsu D, Buchli J (eds) *Robotics: science and systems XI*, Sapienza University of Rome, Rome, Italy, July 13–17, 2015
- Manning CD, Raghavan P, Schütze H (2008) *Introduction to information retrieval*. Cambridge University Press, New York
- Massimo D, Ricci F (2018) Harnessing a generalised user behaviour model for next-POI recommendation. In: Pera S, Ekstrand MD, Amatrian X, O'Donovan J (eds) *12th ACM conference on recommender systems, RecSys '18*, Vancouver, BC, Canada, October 2–7, 2018. ACM, New York, pp 402–406
- Massimo D, Ricci F (2019) Clustering Users' POIs Visit Trajectories for Next-POI Recommendation. In: Pesonen J, Neidhardt J (eds) *Information and Communication Technologies in Tourism 2019*, ENTER 2019, Nicosia, Cyprus, January 30–February 1, 2019. Springer, pp 3–14
- Mobasher B, Dai H, Luo T, Nakagawa M (2002) Using sequential and non-sequential patterns in predictive web usage mining tasks. In: *Proceedings of the 2002 IEEE international conference on data mining (ICDM 2002)*, 9–12 December 2002, Maebashi City, Japan. IEEE Computer Society, Washington, pp 669–672
- Muntean CI, Nardini FM, Silvestri F, Baraglia R (2015) On learning prediction models for tourists paths. *ACM Transaction on Intelligent Systems and Technology* 7(1):8:1–8:34
- Ng AY, Russell SJ (2000) Algorithms for inverse reinforcement learning. In: Langley P (ed) *Proceedings of the seventeenth international conference on machine learning (ICML 2000)*, Stanford University, Stanford, CA, USA, June 29–July 2, 2000, pp 663–670. Morgan Kaufmann, Burlington
- Palumbo E, Rizzo G, Troncy R, Baralis E (2017) Predicting your next stop-over from location-based social network data with recurrent neural networks. In: Neidhardt J, Fesenmaier DR, Kuflik T, Wörndl W (eds) *Proceedings of the 2nd workshop on recommenders in tourism co-located with 11th ACM conference on recommender systems (RecSys 2017)*, Como, Italy, August 27, 2017. CEUR workshop proceedings, vol 1906, pp 1–8
- Park Y-J, Tuzhilin A (2008) The long tail of recommender systems and how to leverage it. In: *Proceedings of the 2008 ACM conference on recommender systems, RecSys '08*, New York, NY, USA, 23–25 October 2008. Association for Computing Machinery, New York, pp 11–18
- Perentis C, Vescovi M, Lepri B (2015) Investigating factors affecting personal data disclosure. In: *Proceedings of the 24th international conference on world wide web, WWW '15 Companion*, New York, NY, USA, 2015. ACM, New York, pp 89–90
- Poikela M, Schmidt R, Wechsung I, Möller S (2014) Locate!—when do users disclose location? In: *Workshop on privacy personas and segmentation (PPS) at the tenth symposium on usable privacy and security (SOUPS)*, Menlo Park, CA, July 9–11, 2014
- Rajaraman A, Ullman JD (2011) *Data mining*. Cambridge University Press, Cambridge, pp 1–17
- Revfine.com (2019) Travel and tourism industry; an complete overview of all activities. *Travel and Tourism Industry; An complete Overview of All Activities*. www.revfine.com/travel-and-tourism/. Accessed 5 May 2019
- Ricci F, Rokach L, Shapira B (2015) Recommender systems: introduction and challenges. In: Ricci F, Rokach L, Shapira B (eds) *Recommender systems handbook*. Springer US, Boston, pp 1–34
- Sanchez P, Bellogín A (2021) On the effects of aggregation strategies for different groups of users in venue recommendation. *Information Processing and Management* 58:102609
- Smith HJ, Milberg SJ, Burke SJ (1996) Information privacy: measuring individuals' concerns about organizational practices. *MIS Quarterly: Management Information Systems* 20(2):167–196
- Sutton RS, Barto AG (1998) *Reinforcement learning—an introduction*. Adaptive computation and machine learning. MIT Press, Cambridge
- Suzuki S (2018) Comparative analysis of human movement prediction: space syntax and inverse reinforcement learning. [arXiv:1801.00464](https://arxiv.org/abs/1801.00464) [cs.MA]
- Torrijos S, Bellogín A, Sánchez P (2020) Discovering related users in location-based social networks. In: Kuflik T, Torre I, Burke R, Gena C (eds) *28th ACM conference on user modeling, adaptation and personalization, UMAP 2020*, Genoa, Italy, July 12–18, 2020. ACM, New York, pp 353–357

- Wang H, Shen H, Ouyang W, Cheng X (2018) Exploiting POI-specific geographical influence for point-of-interest recommendation. In: Lang J (ed) 27th International joint conference on artificial intelligence, IJCAI 2018, Stockholm, Sweden, July 13–19, 2018. IJCAI, pp 3877–3883
- World Tourism Organization (1995) Collection of tourism expenditure statistics. World Tourism Organization (UNWTO), Madrid
- Yang L, Cui Y, Xuan Y, Wang C, Belongie S, Estrin D (2018) Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In: Proceedings of the 12th ACM conference on recommender systems, RecSys '18, New York, NY, USA, 2018. Association for Computing Machinery, New York, pp 279–287
- Zhang C, Zhang H, Wang J (2018) Personalized restaurant recommendation method combining group correlations and customer preferences. *Information Sciences* 454–455:128–143
- Ziebart BD, Maas AL, Bagnell JA, Dey AK (2008) Maximum entropy inverse reinforcement learning. In: Fox D, Gomes CP (eds) Proceedings of the twenty-third AAAI conference on artificial intelligence, AAAI 2008, Chicago, Illinois, USA, July 13–17, 2008. AAAI Press, Palo Alto, pp 1433–1438

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

David Massimo¹ · Francesco Ricci¹

Francesco Ricci
fricci@unibz.it

¹ Free University of Bozen-Bolzano, Bolzano, Italy