Users' Evaluation of Next-POI Recommendations

David Massimo
damassimo@unibz.it
Free University of Bolzano
Italy

Francesco Ricci
fricci@unibz.it
Free University of Bolzano
Italy

ABSTRACT

The performance of a Recommender System (RS) is often assessed offline, by measuring the system accuracy in predicting or reconstructing the observed user ratings or choices. As a consequence, RSs optimised for that performance measure may suggest items that the user would evaluate correct but uninteresting, because lacking novelty. In fact, these systems are hardly able to generalise the preferences directly derived from the user's observed behaviour. To overcome this problem a novel RS approach has been proposed. It applies clustering to users' observed sequences of choices in order to identify like-behaving users and to learn a user behavioural model for each cluster. It then leverages the learned behaviour model to generate novel and relevant recommendations, not directly the users' predicted choices. In this paper we assess in a live user study how users evaluate recommendations produced by more traditional approaches and the proposed one along different dimensions. The obtained results illustrate the differences of the compared approaches, the benefits and the limitations of the proposed RS.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Human-centered computing** → **User studies**.

KEYWORDS

recommender systems, inverse reinforcement learning, clustering, user study

1 INTRODUCTION

The tourism industry grounds on fulfilling the needs, e.g., accommodation and transportation, of people when moving to a place, for leisure or business purposes [14]. In this industry companies offer online to tourists a wide spectrum of services and activities, such as, city tours, accommodations and food services [15]. However, often the set of available options is so rich that choosing suitable ones can be overwhelming. In order to address this problem, ICT practitioners and far-sighted industries started to develop and employ ad-hoc RSs techniques. Nowadays, the business of companies such as Expedia[1], Booking[2] and Kayak[3] is rooted on recommendation technologies.

In fact, recommender systems are software tools that aim at easing human decision making [16]. In the tourism domain some special dimensions of the recommendation process play an important role. First of all, the demand of activities that a tourist may ask varies in the type and quantity in different contexts. For instance, a tourist may prefer to relax in a park on a sunny day while to visit a museum when it is raining. In order to address this type of requests, Context-Aware RSs (CARS) have been developed [1]. Moreover, since individuals typically consume more than a service or perform more than one activity in a single visit to a destination, session- and sequence-aware recommender systems have been introduced [10]. In tourism applications these methods are used to implement next-POI (Point of Interest) recommendation: recommendations for significant places that the user may be interested to visit next, i.e., after she has visited already some other places (the same day or previously).

In a previous research we developed a novel context-aware recommendation technology (here called *Q-BASE*) for suggesting a sequence of items after the users has already experienced some of them. It models with a reward function the "satisfaction" that a Point of Interest, with some given features, provides to a user [8, 9]. This technique learns the reward function by using only the observation of the users' sequences of visited POIs. This is an important advantage, since typically in on-line systems users scarcely provide feedback on the used services or the visited places. The reward function is estimated by Inverse Reinforcement Learning (IRL), a behaviour learning approach that is widely used in automation and behavioural economics [3, 5]. Moreover, since it is hard to have at disposal the full knowledge, or a huge part of the user history of travel related choices, which would be needed to learn the reward function of a single individual, in [8, 9] IRL is instead applied to clusters of users, and a single learned reward function is therefore shared by all the users in a cluster. For this reason we say that the system has learned a generalised, one per cluster, tourist behaviour model, which identifies the action (POI visit) that a user in a cluster should try next. We studied the proposed approach and compared it with popular baseline algorithms for next-item recommendation [4, 7]. In an offline analysis we have shown that a session-based nearest neighbour algorithm (*SKNN*) generates more precise recommendations while *Q-BASE*, our technique, suggests POIs that are more novel and higher in reward. Hence, we conjectured that, in a real scenario, the latter recommendations may be more satisfying for the user.

In this paper we want to verify that hypothesis, i.e, that users like more the recommendations produced by *Q-BASE*. Moreover, we conjecture that an important difference between the *Q-BASE* method and those based on *SKNN* relies in the "popularity bias": *SKNN* tends to recommend items that have been chosen often by the observed users, while *Q-BASE* is not influenced directly by the popularity of the items, but rather by the popularity of their features. Hence, we introduce here two novel hybrid algorithms that are based on *Q-BASE*, but they deviate from *Q-BASE* by using a popularity score: more popular items tend to be recommended more. These two hybrid algorithms are called *Q-POP COMBINED*

[1] www.expedia.com
[2] www.booking.com
[3] www.kayak.com

and *Q-POP PUSH*. They both combine (in a slightly different way) the item score derived from the reward of the item with a score derived from the item popularity in the users' behaviour data set: more often chosen items (popular) receive a larger score. The items with the largest combined scores are recommended.

We have here repeated the offline analysis of the original *Q-BASE* algorithm and compared its performance with the performance of the two above-mentioned hybrid algorithms, and of two kNN algorithms: *SKNN* that recommends next-item to a user by considering her current session (e.g., visit trajectory) and seeking for similar sessions in the dataset; and sequential session-based kNN (*s-SKNN*) that leverages a linear decay function to weight more in the prediction formula the neighbor trajectories that contain the user's last selected item. Repeating the offline analysis was necessary to validate the conjecture that a significant performance difference between *Q-BASE*- and the *SKNN*- based models is due to the popularity bias of KNN methods. We measure the algorithms offline performance in terms of reward, precision and novelty as it was done in [8]. Moreover, we investigate the effect of the above mentioned hybridization of *Q-BASE*; whether this approach can generate recommendations similar to those computed by *SKNN*. To this end, we compare the Jaccard similarity, of the recommendations (sets) produced by Q-BASE and the hybrid variants, with the recommendations produced by SKNN.

The results of the *offline evaluation* confirm our conjecture: hybridizing *Q-BASE* with item popularity, although it reduces novelty, it increases (offline) precision, aproaching the precision of *SKNN*. Moreover, we show that *Q-POP COMBINED* can still achieve a high reward, whereas *Q-POP PUSH* looses some reward but obtains the same precision of *SKNN*. It is worth noting that as the precision of the proposed hybrid models increase, more and more their produced recommendations overlap with those generated by *SKNN*.

The second major contribution discussed in this paper is an interactive online system aimed at assessing with real users the novelty of and the user satisfaction for the recommendations generated by: the original *Q-BASE* model, one of the two hybrid models (*Q-POP PUSH*) and the same *SKNN* baseline used in the previously conducted offline studies. In the online system the users can enter the set of POI that they previously visited (in Florence) and can receive suggestions for next POIs to visit.

By analysing the users evaluations of the POIs recommended in the *online test*, we found a confirmation that *Q-BASE* suggests more novel items while *SKNN*, as well as the proposed hybrid model *Q-POP PUSH*, offers suggestions that the users like more. We conjecture that, since many items suggested by *Q-BASE* are novel for the users, they are difficult to be evaluated (and liked). We further analyse this aspect by considering recommended items that have been evaluated as "liked and novel" by the users. The results show that *Q-BASE* is better than *SKNN* and *Q-POP PUSH* in suggesting novel and relevant items, which we believe is the primary goal of a recommender system.

In conclusion, in this paper we extend the state of the art in next-POI recommender system with the following contributions:

- Two novel models, *Q-POP COMBINED* and *Q-POP PUSH*, that hybridize the IRL model presented in [8] with a score derived from item popularity.
- An offline study where we show that the proposed hybrid models can obtain precisions similar to those obtained by *SKNN* and *s-SKNN*.
- In a user study we show that when the precision of an algorithms is estimated by leveraging the real user feedback as ground truth, rather than by using the standard ML fictional splitting of train/test, *Q-BASE* performs better than *SKNN* and *Q-POP PUSH* in recommending *novel* items that are liked by the user but it is not better in recommending generic items that are liked.

The paper structure is as follows. In Section 2 the most related works are presented. Then, Section 3 describes how the original IRL-based recommendations are generated [8] and introduces two IRL-based hybrid models. Then, we show how the proposed algorithms compares offline against: the original IRL-based model and the KNN baselines. Section 5 introduces the system developed for the user evaluation and the evaluation procedure. Then, we present the evaluation results. Finally, in Section 7 the conclusion and future works of this study are discussed.

2 RELATED WORK

Our research is focussed on behaviour learning and recommender systems that leverage such behaviour models. Our application scenario is tourism: the goal is to support tourists in identifying what POI they could visit next, given their current location and the information about their past visited places.

Processing and analysing sequences of actions in order to understand the user behaviour to support human decision-making has been already explored in previous research. In [10] is proposed a framework for online experience personalization that leverages users interactions (e.g., clicks) in the form of a sequence. The approach is based on pattern mining techniques in order to identify candidate items, which are present in other users' sequences, that are suitable for recommendations. Another pattern-discovery approach applied to tourism is presented in [13]. Here, the authors propose a RS that identifies next-POI to visit relying on users' check-in sequences data. At first, a directed graph is built from the check-in data and then it is used to identify neighbours of a target user given her check-in data. When neighbours are identified, the POIs in their check-in data are scored. The recommended POI is the one with the maximal score.

Other, more general, pattern-discovery methods are described in [4, 7]. Here the authors present nearest neighbour RS approaches that leverage user behaviour logs: session-based KNN (SKNN) and sequence-aware SKNN (s-SKNN). SKNN seeks for similar users in the system stored logs and identifies the next-item to be recommended given the current user log (session). The s-SKNN weights more weight the neighbours sessions containing the most recent (observed) items of the target user sequence. These methods have been applied to different next-item recommendation tasks showing good performance.

The common aspect of pattern-discovery approaches is that they extract common patterns from user behaviour logs and then learn a predictive model for the next most likely observed user action. That said, these approaches are opaque in explaining the predicted

user behaviour, i.e., users' preferences and their action-selection policy.

To fulfil the need of learning an explainable user behavioural model imitation learning is a viable solution. It is typically addressed by solving Markov Decision Problems (MDP) via Inverse Reinforcement Learning (IRL)[12]. Given a demonstrated behaviour (e.g., user actions sequences) IRL models solve the target MDP by computing a reward (utility) function that makes the behaviour induced by a policy (the learning objective) close to the demonstrated behaviour. In [21] the authors developed an IRL approach based on the principle of maximum entropy that is applied in the scenario of road navigation. The approach is based on a probabilistic method that identifies a choice distribution over decision sequences (i.e., driving decisions) that matches the reward obtained by the demonstrated behaviour. This technique is useful to model route preferences as well as to infer destinations based on partial trajectories. In [3] the authors propose an IRL-based solution to the problem of learning a user behaviour at scale. The application scenario is migratory pastoralism, where learning involves spatio-temporal preferences and the target reward function represents the net income of the economic activity. Similarly, in [5] it is proposed a method for computing the reward humans get by their movements decisions. The paper presents a tractable econometric model of optimal migration, focusing on expected income as the main economic influence on migration. The model covers optimal sequences of location decisions and allows for many alternative location choices. All these works, focus on designing a choice model without studying their application to RSs.

In this work we present two variants of the IRL-based recommender system presented in [8]. There is proposed a RS that first learns users behaviour via IRL and then harnesses it to generate next-item recommendations. In an offline evaluation we showed that the approach excels in novelty and reward, whereas, more precise recommendations are generated by *SKNN-based* techniques. In this paper we argue that the ability of pattern-discovery methods to score high in precision is related to the fact that they are discriminative and are influenced by the observed popularity of the items in the training data. Therefore, in order to leverage item popularity also in an IRL model, we extend the it by hybridizing its scoring function (Q function) with item popularity.

3 RECOMMENDATION TECHNIQUES

3.1 User Behavior Modelling

In this paper, user (tourist) behaviour modelling is based on Markov Decision Processes (MDP). A MDP is defined by a tuple (S, A, T, r, γ). S is the state space and, in our scenario, a state models the visit to a POI in a specific context. The contextual dimensions are: the weather (visiting a POI during a sunny, rainy or windy time); the day time (morning, afternoon or evening); and the visit temperature conditions (warm or cold). A is the action space; in our case it represents the decisions to move to a POI. Hence, POIs and actions are in biunivocal relation. A user that is in a specific POI and context can reach all the other POIs in a new context. T is a finite set of probabilities. $T(s'|s, a)$ is the probability to make a transition from state s to s' when action a is performed. For example, a user that visits Museo del Bargello in a sunny morning (state s_1) and wants to visit Giardino di Boboli (action a_1) in the afternoon can arrive to the desired POI with either a rainy weather (state s_2) or a clear sky (state s_3). The transition probabilities may be equal, $T(s_2, a_1|s_1) = 0.5$ and $T(s_3, a_1|s1) = 0.5$. The function $r : S \rightarrow \mathbb{R}$ models the reward a user obtains from visiting a state. This function is unknown and must be learnt. We take the restrictive assumption that we do not know the reward the user receives from visiting a POI (the user is not supposed to reveal it). But, we assume that if the user visits a POI and not another (nearby) one then this signals that the first POI gives her a larger reward than the second. Finally, $\gamma \in [0, 1]$ is used to measure how future rewards are discounted with respect to immediate ones.

3.2 User Behavior Learning

Given a MDP, our goal is to find a policy $\pi^* : S \rightarrow A$ that maximises the cumulative reward that the decision maker obtains by acting according to π^* (optimal policy). The value of taking a specific action a in state s under the policy π, is computed as $Q_\pi(s, a) = \mathbf{E}^{s,a,\pi}[\sum_{k=0}^{\infty} \gamma^k r(s_k)]$, i.e., it is the expected discounted cumulative reward obtained from a in state s and then following the policy π. The optimal policy π^* dictates to a user in state s to perform the action that maximizes Q. The problem of computing the optimal policy for a MDP is solved by reinforcement learning algorithms [18].

We denote with ζ_u a user u trajectory, which is a temporally ordered list of states (POI-visits). For instance, $\zeta_{u_1} = (s_{10}, s_5, s_{15})$ represent a user u_1 trajectory starting from state s_{10}, moving to s_5 and ending to s_{15}. With Z we represent the set of all the observed users' trajectories which can be used to estimate the probabilities $T(s'|s, a)$.

Since, typically users of a recommender system scarcely provide feedback on the consumed items (visited POIs), the reward a user gets by consuming an item is not known. Therefore, the MDP, which is essential to compute the user policy, cannot be solved by standard Reinforcement Learning techniques. Instead, by having at disposal only the set of POI-visit observations of a user (i.e., the users' trajectories), a MDP for each user could be solved via Inverse Reinforcement Learning (IRL) [12]. In particular, IRL enables to learn a reward function whose optimal policy (the learning objective) dictates actions close to the demonstrated behavior (the user trajectory). In this work we have used Maximum likelihood IRL [2].

3.3 Clustering Users with Similar Behavior

Having the knowledge of the full user history of travel related choices, which would be needed to learn the reward function of a single individual, is generally hard to obtain. Therefore, IRL is here applied to clusters of users (trajectories) [8, 9]. This allows to learn a reward function that is shared by all the users in a cluster. Hence, we say that the system has learned a generalized tourist behavior model, which identifies the action (POI visit) that a user in a cluster should try next.

Clustering the users' trajectories is done by grouping them according to a common semantic structure that can explain the resulting clusters. This is accomplished by employing Non Negative Matrix Factorization (NMF) [6]. NMF extracts topics, i.e., lists of words, that describe groups of documents. Therefore, in order to

apply NMF, we build a document-like representation of a user trajectory that is based on the features (terms) that describe the states visited in a trajectory. Hence, a document-like representation is build for each trajectory in the set Z.

3.4 Recommending Next-POI visits

Here we propose two new next-POI recommendations techniques, *Q-POP COMBINED* and *Q-POP PUSH*, that extend the pure IRL-based *Q-BASE* model, already introduced in [8] (where it was called CBR).

Q-BASE. The behavior model of the cluster the user belongs to is used to suggest the optimal action this user should take next, after the last visited POI. The optimal action is the action with the highest Q value in the user current state [8].

Q-POP COMBINED. In order to recommend more popular items, we propose to hybridise the generalized tourist behavior model learnt for the cluster to which the user belongs to with the item popularity. In particular, given the current state s of a user, for each possible POI-visit action a that the user can make, we apply the following transformation $Q'(s,a) = \frac{Q(s,a)}{\Sigma_i^{|A|} Q(s,a_i)}$ and then we multiply $Q'(s,a)$ by the probability that a POI appears in a user trajectory (in a given data set Z). The result of the multiplication is a distribution that is used to sample the next-POI visit action recommended to the user.

Sampling from a distribution derived from functions composition is widely done in simulation [17]. The approach tries to simulate the decision making process of a user that has all the elements to decide how to act next, i.e., she knows the reward of her future action (the Q values), but she is also biased to select popular items. We conjecture that this method recommends more popular items that have a large reward as well.

Q-POP PUSH. The second hybrid recommendation method introduces even a higher popularity bias to the recommendations generated by *Q-BASE*. We conjecture that it can obtain even a better precision than *Q-POP COMBINED*, closer to the precision of the *SKNN*-based methods. *Q-POP PUSH* scores the visit action a in state s as following:

$$score(s,a) = (1+\beta^2)\frac{Q(s,a)\cdot pop(a)}{(Q(s,a) + pop(a)\cdot\beta^2)}$$

This is the harmonic mean of $Q(s,a)$ and $pop(a)$, which is the scaled (i.e., min-max scaling) counts $c_Z(a)$ (in the data set Z) of the occurrences of the POI-visit corresponding to action a. The harmonic mean is widely used in information retrieval to compute the F1-score. In our case the parameter β was set to 1. The action recommended to the user is the one with the highest score.

4 OFF-LINE ALGORITHM ANALYSIS

4.1 Baselines

We compare here the performance of the recommendations generated by the above mentioned methods with two nearest neighbor baselines: *SKNN* and *s-SKNN*.

SKNN [4] recommends the next-item (visit action) to a user by considering her current session (trajectory) and seeking for similar sessions (neighbourhood) in the data-set. The neighbourhood, i.e., the closest trajectories to the current trajectory, are obtained by computing the binary cosine similarity between the current trajectory ζ and those in the dataset ζ_i: $c(\zeta,\zeta_i)$. Given a set of nearest neighbours N_ζ the score of a visit action a can be computed as:

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

With 1_{ζ_n} we denote the indicator function: it is 1 if the POI selected by action a appears in the neighbour trajectory ζ_n (0, otherwise). In our data set we cross validated the optimal number of neighbours, and this number is close to the full cardinality of the data set. The recommended actions are those with the highest scores.

s-SKNN [7] extends *SKNN* by employing a linear decay function w_ζ to weight more in the prediction formula the neighbor trajectories that contain the user's last observed visit action and less the earlier visits. The current user trajectory's neighborhood is obtained as in *SKNN*, while the computation of the score of a visit action is as following:

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

For instance, let us say that a_3 is the third observed visit action in the user trajectory ζ (where $|\zeta| = 5$) and that a_3 appears in the trajectory $\zeta_n \in N_\zeta$, then the weight defined by the decay function is $w_{\zeta_n} = 3/5$. Also for *s-SKNN*, the recommended actions are those with the highest scores.

4.2 Evaluation Metrics

The evaluation metrics used to assess the algorithm performance are reward, as defined in [8], precision, novelty and recommendations similarity. Let us denote with $Rec_{u,s}$ a list of recommendations for the user u in state s, and a_o the observed (next) POI-visit (test item). **Reward** measures the average increase in reward that the recommended actions give compared to the observed one:

$$reward(Rec_{u,s}, a_o) = (\sum_{a \in Rec_{u,s}} Q(s,a) - Q(s,a_o))/|Rec_{u,s}|$$

Novelty estimates how unpopular are the recommended visit actions and ranges in $[0,1]$. A POI is assumed to be unpopular if its visits count is lower than the median of this variable in the training set. Let U be the set of unpopular POIs and $1_U(a)$ its indicator function (it is 1 if $a \in U$ and 0 otherwise), novelty is defined as follows:

$$novelty(Rec_{u,s}) = \frac{\sum_{a \in Rec_{u,s}} 1_U(a)}{|Rec_{u,s}|}$$

Let obs_u be the set of observed POI-visit actions in the user u trajectory (test set). The indicator function $1_{obs_u}(a)$ is 1 if $a \in obs_u$ and 0 otherwise. **Precision** is then computed as follows:

$$precision(Rec_{u,s}) = (\sum_{a \in Rec_{u,s}} 1_{obs_u}(a))/|Rec_{u,s}|$$

Finally, we estimate the **Similarity** of two lists of recommendations by computing their Jaccard index. In this study, we compute

the Jaccard index of the recommendations generated by our proposed methods and those generated by *SKNN*. The goal is to verify whether the proposed hybrid methods, which recommend more popular items, improve some of the performances of the pure IRL method Q-BASE and if they recommend items more similar to those recommended by *SKNN*.

4.3 Off-line Study Results

In this study we used an extended version of the POI-visit data-set presented in [11]. It consists of tourist trajectories reconstructed from the public photo albums of users of the Flickr[4] platform. From the information about the GPS position and time of each single photo in an album the corresponding Wikipedia page is queried (geo query) in order to identify the name of the related POI. The time information is used to order the POI sequence derived from an album. In [9] the dataset has been extended by adding information about the context of the visit (weather summary, temperature and part of the day), as well as POI content information (historic period of the POI, POI type and related public figure). In this paper we used an extended version of the dataset that contains 1668 trajectories and 793 POIs.

Trajectories clustering identified 5 different clusters, as in the previous study. In Table 1 we report the performances of Top-1 and Top-5 recommendations for the considered methods. We immediately observe that *SKNN* scores higher in precision, whereas *Q-BASE* suggests more novel and with higher reward items. These results confirm previous analysis [8, 9]. *SKNN* and *s-SKNN* perform very similarly, hence, in this data-set, the sequence-aware extension of *SKNN* seems not to offer any advantage.

When comparing *Q-POP COMBINED* and *Q-POP PUSH* with the two *SKNN*-based methods we found that *Q-POP COMBINED* has a good trade-off between reward and precision. In particular, reward is 4 times (Top-1) the reward of both *SKNN* and *s-SKNN* while precision increases considerably with respect to *Q-BASE*. The same is observed for Top-5 recommendations. But novelty is penalised by the popularity bias of this method.

By looking at the performance of *Q-POP PUSH* we can confirm our study conjecture: a stronger popularity bias enables the algorithm to generate recommendations that are more precise and in particular the precision of *Q-POP PUSH* is equal to that of *SKNN* and *s-SKNN*. But, as expected, reward and novelty are penalised.

With regard to the similarity (Jaccard index) of the recommendations generated by the proposed methods with those of *SKNN*, we can clearly see that the more the precision increases, the higher the Jaccard index becomes. So, the methods are more precise as they are more similar to *SKNN*.

5 ONLINE USER EVALUATION

We conducted an online user-study in order to measure the users' perceived novelty and satisfaction for the recommendations generated by the *Q-BASE* model, the hybrid model *Q-POP PUSH* and the *SKNN* baseline used in the offline study. We designed an online system which first profiles the user by asking her to enter as many as possible previously visited POIs (in Florence). Then the user is asked to evaluate a list of recommendations generated by the aforementioned three models, without being informed of which algorithm recommends what. The data used by the system to train the models and compute recommendations is the same of the offline study, a catalogue of 793 items.

[4] www.flickr.com

Table 1: Recommendation performance

Models	Q-BASE	Q-POP C	Q-POP P	SKNN	s-SKNN
Rew@1	0.073	0.023	-0.002	-0.007	-0.009
Prec@1	0.043	0.057	0.099	0.109	0.109
Nov@1	0.061	0.029	0.000	0.000	0.000
Jacc@1	0.085	0.106	0.424	-	0.791
Rew@5	0.032	0.017	-0.009	-0.010	-0.010
Prec@5	0.045	0.049	0.060	0.068	0.063
Nov@5	0.122	0.040	0.000	0.000	0.000
Jacc@5	0.061	0.063	0.192	-	0.530

5.1 Online Evaluation System

The interaction with the system unfolds as follow: landing phase; introduction to the experiment and start up questions; preference elicitation phase; recommendation generation and evaluation.

Once the user accesses the website she can select the language (Italian or English) and then, if the user accepts to participate to the experiment, she is asked whether has already been in Florence. If she replies "no" the procedure ends. Otherwise, the user is considered to have some experience of the city and can declare which POIs has already visited. In this case, the preference elicitation phase is supported by a user interface (Figure 1) that enables the user to select as many POIs she remembers to have visited in Florence. The selection can be performed in two non-exclusive modalities. The first one is a lookup bar with auto-completion, while the second is a selection pane that contains the most popular 50 POIs. If the user hovers or taps on an item the system renders a media card presenting content extracted from Wikipedia: a picture and a textual description. When the user selects a POI as visited, this is added to an (editable) list. The selected POIs are meant to build a user profile which is then used to identify the best representative user's trajectory cluster, among the 5 clusters of previously collected training data (the details of this computation are explained in the next section).

Then the system generates a short itinerary (5 POIs) composed by a small sample of the POIs that the users previously declared to have visited (Figure 2). This is the itinerary that the user is supposed to have followed just before asking a recommendation for a new point to visit. We decided to generate a fictitious itinerary because we did not want to ask the user to remember any previous visit itinerary, but we also tried to generate a trajectory that is likely to have been followed (by sampling among the POIs that entered in the profiling stage). By showing a hypothetical itinerary to the user, followed up to the current point, we wanted to reinforce in the user the specific setting of the supported recommendation task: next-POI recommendation.

That said, the recommendation generation and evaluation phase present a user interface that is organized as follows. At the top of the page there is an information box containing the (previously mentioned) hypothetical (5-POIs) trajectory that the user should assume has followed (Figure 2). Below, there is an info box that explains the participant to assume that she has visited the selected

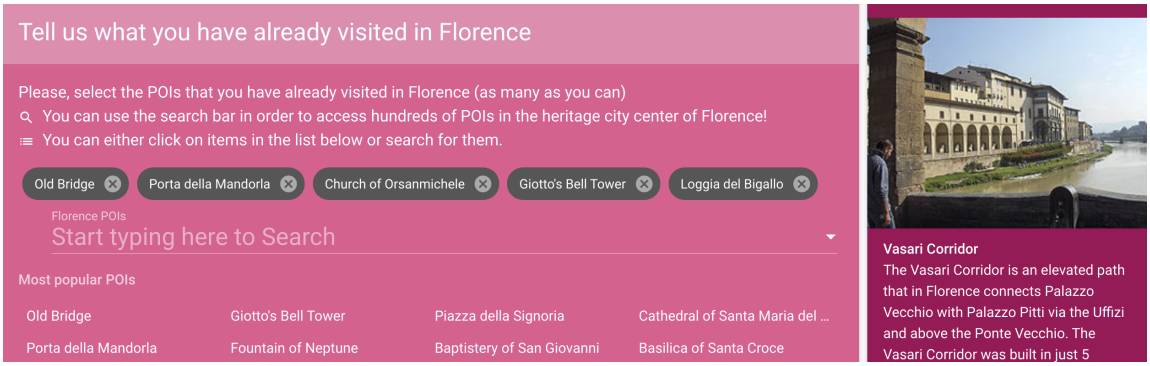

Figure 1: POI selection UI detail.

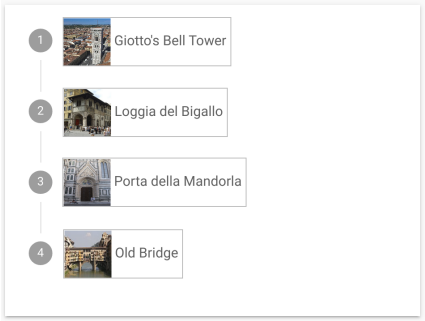

Figure 2: Itinerary detail.

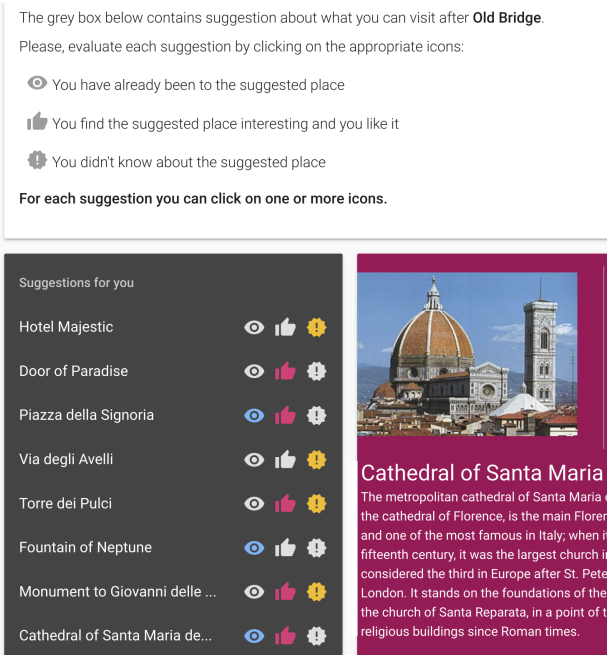

Figure 3: Evaluation. UI detail.

attractions, in the presented order. Finally, the participant is informed that the beneath box (Figure 3) contains a list of POIs that she can visit (recommendations) after the last POI in the itinerary. The user is asked to mark the recommendations 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).

We recruited the experiment participants via social media and mailing lists and we collected over 300 responses of which 202 are from users that visited Florence. After excluding unreliable replies (e.g., survey completed in less than 2 minutes) we counted 158 users. The number of recommended next-POI visits shown to the users is 1119 (approximately three by each of the three methods per user, excluding the items recommended by two or more method simultaneously). Hence on average a user has seen 7.1 recommendations.

5.2 Recommendation List Generation

In order to generate recommendations using *Q-BASE* and *Q-POP PUSH* an online user must be associated to one of the five existing trajectories' clusters. In fact, the user behavioural model is considered to be shared with the other users in the same cluster, and it is learned by using the trajectories already present in the cluster.

Matching a user to a cluster. In order to associate an online user to a pre-existent cluster (among the 5 that we created) we built a tf-idf representation of the POIs (documents) that are in the user profile and then we run a nearest neighbor classifier where the training data are the existent trajectories in the data set, already classified in the 5 clusters. We assessed the classifier performance by splitting the trajectories data set: 80% of the dataset has been used for training the classifier and the remaining 20% has been used as test set. In a 10-fold cross-validation the classifier showed an accuracy of 0.67. Hence, the quality of this classifier is not very high. This may have penalised both *Q-BASE* and *Q-POP PUSH* in the online study.

5 POIS Fictitious Itinerary. Once the user is associated to a cluster, among all the trajectories in the cluster we identify the trajectory in the cluster with the highest overlap (intersection) with the POIs selected by the study participant (randomly breaking ties). On

the user interface, as we mentioned above, in order to avoid information overload, we show to the user at most 5 items, of her user profile, ordered according to the matched itinerary, found in the matched cluster. The itinerary is shown to the user as her current (hypothesized) sequence of visited POIs in order to evaluate the next-POI recommendations as appropriate or not to complete the initiated itinerary.

Recommendations. Given the fictitious hypothesized itinerary followed by the user so far, next-POI recommendations are independently generated leveraging the algorithms *Q-BASE*, *Q-POP* and *kNN*. Then, from the recommendations generated by the algorithms we filter out (post-filtering) the POIs already in the user profile. This is an important feature of our study: we wanted to suggest POIs that are good for a next visit, i.e., that the user has not yet visited [5]. Moreover, in order to avoid biases in the recommendation evaluation phase we do not reveal to the user which recommendation algorithm has produced which POI recommendation.

Furthermore, to control the "position bias" [19, 20], i.e., the tendency of users to select top positioned items in a list, regardless of their relevance, we aggregate the top-3 suggestions of each algorithm without giving to any algorithm a particular priority. In fact, at first, we (randomly for each user) generate an order that we follow to pick items from the three lists of the top-3 suggestions generated by the three considered algorithms. Then we aggregate the three ranked list by picking up, in turn, the items from the top to the bottom of the sorted lists. For instance, if the generated order is *Q-BASE*, *kNN* and *Q-POP*, then, the aggregated list of recommendations (max length 9) that is shown to the user, contains in the first position the top recommendation of *Q-BASE*, then the top item suggested by *kNN* and then that suggested by *Q-POP*. The same pattern is applied for the remaining positions: the fourth item in the aggregated list is the second best POI suggested by *Q-BASE* and at the fifth and sixth positions are placed the second best POIs suggested by *kNN* and *Q-POP*. In the case a POI is suggested by more than one algorithm, the item is shown only once.

6 RESULTS OF THE ONLINE USER STUDY

The results of the recommendation generation and evaluation phase are shown in Table 2. We show here the probabilities that a user marks as "visited", "novel", "liked" (for a next visit) or both "liked" and "novel" an item recommended by an algorithm. They are computed by dividing the total number of items marked as, visited, liked, novel and both liked and novel, for each algorithm, by the total number of items shown by an algorithm. By construction, each algorithm contributes with 3 recommendations in the aggregated list shown to each user. It is worth stressing that a user marked as "liked" an item that she judged as a good candidate for a next POI visit. Hence, here a "like" is not a generic appreciation of the item, but takes (partially) into account the visit context (what items the user has already visited).

We note that the POIs recommended by *SKNN* and *Q-POP* have the highest probability (24%) that the user has already visited them, and the lowest probability to be considered as novel. *Q-BASE* scores a lower probability that the recommended item be already visited (16%) and the highest probability that the recommended item be novel (52%). This is in line with the offline study where *Q-BASE* excels in recommending novel items.

Considering now the user satisfaction for the recommendations (liked), we conjectured that a high reward of an algorithm measured offline, corresponds to a high perceived satisfaction (likes) measured online. But, by looking at the results in Table 2 we have a different outcome. *Q-BASE*, which has the highest offline reward recommends items that an online user likes with the lowest probability (36%). *Q-POP PUSH* and *SKNN* recommend items that are more likely to be liked by the user (46%).

Another measure of system precision that we computed is the probability that a user likes a novel recommended POI, i.e., a POI that the recommender presented for the first time to the user ("Liked & Novel" in Table 2). We note that this is the primary goal of a recommender system: to enable users to discover items that are interesting for them, not to suggest items that the user likes, but that she is already aware of, or she has already consumed. There is poor utility of such a functionality. In this case, *Q-BASE* (highest reward and lowest precision offline) recommends items that a user will find novel and also like with the highest probability (0.09%), whereas *SKNN* and *Q-POP PUSH* recommends items that the user will find novel and will like with a lower probability(0.08%). We believe that the online computed "Liked & Novel" probability is a better measure of the precision of a RS. In fact, the standard offline estimation of precision, which is computed on the base of an artificial split of the available liked items into train/test is not able to estimate how, not yet experienced items that the recommender suggests may be liked by the user. It is also worth noting the low scores of this metric: it is hard to observe a user that liked a novel item. This aspect is further discussed below.

In order to further study the online user evaluation of the recommended items, we have computed the probability that a user will like recommendations given the fact 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. The results of this analysis are shown in Table 3. The novel POIs recommendations generated by *SKNN* and *Q-POP PUSH* are liked more (20% and 22%) than those produced by *Q-BASE* (17%). We believe that this is because often *Q-BASE* suggests items that are very specific and users may find hard to evaluate them. For instance, *Q-BASE* suggests often "Porta della Mandorla" which is a door of the "Duomo". This POI can be perceived as a niche item and much less attractive than the "Duomo" itself. Moreover, by conducting post-survey activities participants declared that it is difficult to like something that is unknown.

In fact, the probability that a user likes a recommended POI that she has visited tends to be much larger. This probability is 31% and 28% for *Q-POP PUSH* and *SKNN* respectively. Whereas, *Q-BASE* also here performs worse (26%). We think that the performance difference is again due to the fact that both *SKNN* and *Q-POP* tend to recommend popular POIs (easier to judge), whereas *Q-BASE* recommends more "niche" items.

Considering now the probability that a user will like an item that she knows but has not yet visited we see again a similar pattern as before: *Q-POP PUSH* and *SKNN* suggest items that will be liked with a higher probability (81% and 80%) than *Q-BASE* (71%). These

[5] Still some recommendations can be not novel because the user will never declare all the POIS that she visited or she knows in the city.

Table 2: Probability to evaluate a recommendation of an algorithm as visited, novel and liked.

	Q BASE	Q POP	SKNN
Visited	0.165	0.245	0.238
Novel	0.517	0.376	0.371
Liked	0.361	0.464	0.466
Liked & Novel	0.091	0.076	0.082

Table 3: Probability that a user likes a suggested item given that she visited, knew or is unaware of it.

	Q BASE	Q POP	SKNN
P(Liked | Novel)	0.176	0.202	0.222
P(Liked | Visited)	0.256	0.310	0.283
P(Liked | Known & Not Visited)	0.717	0.810	0.806

probabilities are very large. We conjecture that this is because these are popular items that the user has not yet visited. In fact, if we compare the probabilities that a user will like an item given that is novel, visited or known but not yet visited, we see that it is the largest for the latter items (> 70%), it is lower for the visited items (> 26%) and and the lowest for the novel items (< 22%). This reinforce the conjecture that users tends to like items they are familiar with (but they have not yet consumed).

7 CONCLUSION AND FUTURE WORK

In this paper we extend the state of the art in IRL-based next-POI RSs. We started our analysis by hypothesising that users like more the recommendations produced by IRL-models and that the poor offline accuracy of these models, compared to KNN approaches, is due to the total absence of a popularity bias in the recommendation generation. For that reason we designed two new hybrid models that bias the pure IRL-model *Q-BASE* to suggest more popular items: *Q-POP COMBINED* and *Q-POP PUSH*.

We show with an offline experiment that the hybridization of *Q-BASE* results in an increase of precision: *Q-POP PUSH* performs equally to SKNN-based approaches.

With an *online test* we show that the *Q-BASE* model excels in suggesting novel items, whereas *SKNN* and *Q-POP PUSH* suggests items that are "liked" more. We also show that if we consider the combined feedback "liked and novel", i.e., recommendations that are liked and are novel to the user, *Q-BASE* outperforms both *SKNN* and *Q-POP PUSH*. Hence, we show that *Q-BASE* may be able to better accomplish the most important task of a RS for tourism: suggesting relevant POIs that are unknown for a user and also relevant.

We emphasize here that the objective of this research is a next-POI RS that harnesses a generalized tourist behavior model. While in this work we showed the benefits of such a RS through a web-based study we are now conducting a novel user study with real tourists interacting with a system while visiting a destination (South Tyrol)[6].

[6]http://wondervalley.unibz.it
https://beacon.bz.it/wp-6/beaconrecommender/

Another future work direction is the analysis of the users' perception of the recommendations generated by the different algorithms given the possibly different users' knowledge of the target destination.

ACKNOWLEDGMENTS

The research described in this paper was developed in the project Suggesto Market Space in collaboration with Ectrl Solutions and Fondazione Bruno Kessler.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin. 2011. Context-Aware Recommender Systems. In *Recommender Systems Handbook*, F. Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). 217–253.
[2] M. Babes-Vroman, V. Marivate, K. Subramanian, and M. Littman. 2011. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning - ICML'11*. 897–904.
[3] S. Ermon, Y. Xue, R. Toth, B. Dilkina, R. Bernstein, T. Damoulas, P. Clark, S. DeGloria, A. Mude, C. Barrett, and C. P. Gomes. 2015. Learning Large Scale Dynamic Discrete Choice Models of Spatio-Temporal Preferences with Application to Migratory Pastoralism in East Africa. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Pattern*, 644–650.
[4] D. Jannach and L. Lerche. 2017. Leveraging Multi-Dimensional User Models for Personalized Next-Track Music Recommendation. In *Proceedings of the Symposium on Applied Computing - SAC'17*. 1635–1642.
[5] J. Kennan and J. R. Walker. 2011. The Effect of Expected Income on Individual Migration Decisions. *Econometrica* 79, 1 (2011), 211–251.
[6] D. D. Lee and H. S. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.
[7] M. Ludewig and D. Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Model. User-Adapt. Interact.* 28, 4-5 (2018), 331–390.
[8] D. Massimo and F. Ricci. 2018. Harnessing a generalised user behaviour model for next-POI recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*. 402–406.
[9] D. Massimo and F. Ricci. 2019. Clustering Users' POIs Visit Trajectories for Next-POI Recommendation. In *Information and Communication Technologies in Tourism 2019, ENTER 2019, Proceedings of the International Conference in Nicosia, Cyprus, January 30-February 1, 2019*. 3–14.
[10] B. Mobasher, H. Dao, T. Luo, and M. Nakagawa. 2002. Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks. In *Proceedings of the IEEE International Conference on Data Mining - ICDM '02*. 669–672.
[11] C. I. Muntean, F. M. Nardini, F. Silvestri, and R. Baraglia. 2015. On Learning Prediction Models for Tourists Paths. *ACM Transactions on Intelligent Systems and Technology* 7, 1 (2015), 1–34.
[12] A. Ng and S. Russell. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning - ICML '00*. 663–670.
[13] S. Oppokhonov, S. Park, and I. K. E. Ampomah. 2017. Current Location-based Next POI Recommendation. In *Proceedings of the International Conference on Web Intelligence (WI '17)*. ACM, New York, NY, USA, 831–836.
[14] World Tourism Organization. 1995. *Collection of Tourism Expenditure Statistics*. World Tourism Organization (UNWTO).
[15] Revfine.com. 2019. Travel and Tourism Industry; An complete Overview of All Activities. https://www.revfine.com/travel-and-tourism
[16] F. Ricci, L. Rokach, and B. Shapira. 2015. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). 1–34.
[17] C. R. P. Robert and G. Casella. 2010. *Introducing Monte Carlo Methods with R*. EU-Nachrichten, Themenheft, Vol. 30. Springer, New York, NY u.a.
[18] R. S Sutton and A. G. Barto. 2014. *Reinforcement Learning: An Introduction (Second edition, in progress)*. The MIT Press.
[19] E. C. Teppan and M. Zanker. 2015. Decision Biases in Recommender Systems. *Journal of Internet Commerce* 14, 2 (2015), 255–275.
[20] X. Wang, M. Bendersky, D. Metzler, and M. Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 115–124.
[21] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. 2008. Maximum Entropy Inverse Reinforcement Learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - AAAI'08*. 1433–1438.