# Next-POI Recommendations Matching User's Visit Behaviour

David Massimo(✉) and Francesco Ricci

Free University of Bolzano, Bolzano, Italy
{damassimo,fricci}@unibz.it

**Abstract.** We consider the *urban tourism* scenario, which is characterized by limited availability of information about individuals' past behaviour. Our system goal is to identify relevant next Points of Interest (POIs) recommendations. We propose a technique that addresses the domain requirements by using clusters of users' visits trajectories that show similar visit behaviour. Previous analysis clustered visit trajectories by aggregating trajectories that contain similar POIs. We compare our approach with a next-item recommendation state-of-the-art Neighbour-based model. The results show that customizing recommendations for clusters of users' with similar behaviour yields superior performance on different quality dimensions of the recommendation.

**Keywords:** Recommender Systems · Clustering · Behaviour learning

## 1 Introduction

User decision-making in *urban tourism* is affected by a multitude of factors: weather conditions; time at disposal; background knowledge of the places to visit; previously visited places; reputation of a place; and many others. Context-aware [1] and session-based Recommender Systems (RSs) have been proposed to tackle similar settings [10,15]. Our specific goal is to support visitors to identify next points of interest (POIs) to visit that match their interests and the specific context of the visit. We aim at developing techniques that identify new POIs that produce a rewarding overall experience that complete the initiated journey. Hence, the recommendations should be diverse, novel, compelling, and not only accurate [4,14], i.e., matching users' expected behaviour [7].

We conjecture that the above mentioned goals can be attained by better understanding and using previously observed users' visit behaviour. We adopt a theory-driven approach to model not only *what* tourists visits but also *how* they do that, on the base of their conduct in the destination. In particular, we represent a user's POI-visit trajectory, a sequence of POI visits, with four inter-dependent behavioural characteristics that qualify the tourist conduct in an urban destination: *selectivity*, *rapidity*, *repetition* and *capriciousness* [2].

*Selectivity* indicates that tourists consume only a very small portion of what a city has to offer, due to a limited knowledge of the destination, time to perform

W. Wörndl et al. (Eds.): *Information and Communication Technologies in Tourism 2021*, pp. 45–57, 2021.
https://doi.org/10.1007/978-3-030-65785-7_4

the POIs selection/visit and a scarce willingness to move [16]. *Rapidity* relates to the temporal dimension in the visit of a destination; users can be interested to visit few POIs for a longer period while others may want to see as many POIs as possible [5]. *Repetition* indicates that repeated visits are typically superfluous, and users rather extend their visits to novel places that share some characteristics with the visited ones [6]. *Capriciousness* stresses that tourists' choices are influenced by the social context. Tourists often follow touristic trends, being influenced by POIs popularity and fashion [8], which is easily communicated by websites like TripAdvisor.

In this study we operationalize these four behavioral characteristics with five available features (Sect. 3.1) that grasp the essential dimensions of the users' conduct in a city, and we use them to build a low-dimensional representation of the users' POI-visit trajectories. We then identify clusters of users that share *what* and *how* they visit a destination. These clusters are used to learn cluster-specific behaviour models and to generate recommendations. We conjecture that these recommendations are more effective (along several dimensions) than: a) those generated by clustering POI-visits trajectories by solely using POI content features [13] and b) a state of the art session based RS (`SKNN`).

We propose here a novel Inverse Reinforcement Learning based recommendation algorithm called `Q-BASEX` that allows to generate next-POI visit actions recommendations for new POIs (not observed in the training) and even for user states whose contextual conditions have not been observed in the training set. `Q-BASEX` is evaluated on two POI-visit trajectories data sets (Rome and Florence) by measuring: the precision of the recommendations; how they match to the user's expected visit experience; the coverage of suggested relevant items; the diversity of the items suggested to the various users; the recommended items (un)popularity. `Q-BASEX` is compared with a next-item nearest neighbour-based recommendation model Session KNN (`SKNN`) [10] that in previous studies resulted to be more accurate than other Inverse Reinforcement Learning methods [12,13].

The obtained results show that clustering users with similar behaviour allows to better support the visitors of an urban area. In particular, the proposed method is more accurate than the recommendation baselines, and suggests items that are closer to the user's interest and more rewarding.

## 2 Related Work

The exploitation of data describing a sequence of human actions (e.g., choices or web pages visits), in order to support user decision making, has been often studied in the past. Content personalization by leveraging users' interaction sequences on the web (e.g., e-commerce) has been studied in [15]. Here, by mining data, recording the sequence of items a user interacted with, a set of candidate items is found in order to generate recommendations to similar users. Other RSs approaches that leverage users' behavioural data are based on nearest neighbour and neural network methods. Session KNN (`SKNN`) is a nearest neighbour-based RSs approach that exploit users' behavioural data logs that are similar (neighbours) to the logs of a target user. `GRU4REC` is another popular method used in session-based RSs. It uses a Gated Recurrent Unit (GRU) Recurrent Neural Network and predicts the next action (i.e., next item to purchase) of a target user given information on her past action sequences.

Other approaches are based on Reinforcement Learning and generate next-action recommendations by estimating the user's reward obtained from a sequence of (optimal) choices [17,20]. Here, the reward function is known, i.e., users provide feedback for the consumed items, which is not always the case in reality. Hence, in order to learn an explainable user behavioural model, without relying on explicit user feedback, Inverse Reinforcement Learning (IRL) has been used [19]. IRL models estimate the reward function that makes the behaviour, induced by the optimal policy of the estimated reward function, close to the observed data. In [12,13] IRL was used to generate next-POIs recommendations.

RSs are designed and evaluated predominately by measuring the recommendation accuracy, which relates to the ability to correctly predict the observed user choices [7,11]. However, it has been pointed out that optimizing a RS for accuracy yields suggestions that are seen by the user as obvious and repetitive: they too closely match what the user is normally doing. In fact, in [4,14] it is argued that a proper assessment of a RS should be based on a wider spectrum of metrics.

In [9] tourists' collective information about their activities in a city is used to identify POIs of interest and the tourists' behaviour in an urban area. The authors employ a density based clustering algorithm (POI identification) and association-rule mining (behaviour analysis) on users' geo-localized photos uploaded on a photo sharing platform. They propose this approach in order to identify POIs to recommend to a user. In that work collective users' information is aggregated, hence losing the information about the sequence of decisions.

The generation of recommendations using user's sequences of behavioural data is also discussed in [21]. The authors revisit the trajectory clustering problem [22], which generally leverages spatio-temporal similarity measures, in order detect clusters of trajectories in different regions and time periods. This is achieved by learning a low-dimensional representations of the trajectories. As in our approach they reduce the dimensionality of a trajectory by discarding space and time features at the POI-visit level. But, they also discard the global temporal aspect of a trajectory that instead we consider among the characterizing factors of the users' behaviour.

## 3 Next-POI Recommendation with QBASEX

### 3.1 Data

We have analysed two data sets of geo-localized POI-visits trajectories, recorded via GPS sensors in the cities of Florence and Rome (Italy) [18]. Each trajectory describes the successive visits of a tourist in one day in one city. The total visit time of a trajectory spans from 30 min to the whole day.

We represent each POI-visit in terms of its content and context, namely: an hourly weather summary (e.g., cloudy), temperature (e.g., cold) and daytime (e.g., evening). We use a weather API and the recorded user's stay points to obtain that information. Moreover, from TripAdvisor data we match GPS locations to POIs that were likely visited by the tourists and determine POIs' categories (e.g., museum). In addition, we extract "expert" knowledge from TripAdvisor crowd-sourced data, indicating the POI reputation (e.g., ratings).

The total number of distinct POIs, POI-visit trajectories and features in the above mentioned categories are shown in Table 1.

**Table 1.** Rome and Florence POIs data sets global characteristics (Source: authors)

| Dataset | # POIs | # Trajectories | # Features | | | |
|---|---|---|---|---|---|---|
| | | | Context | Content | "Expert" | Behaviour |
| Florence | 316 | 2110 | 15 | 29 | 9 | 5 |
| Rome | 376 | 4340 | 14 | 28 | 9 | 5 |

In addition to the above mentioned features, we heuristically identified in the available data five hand-crafted features that operationalise the behavioural dimensions that characterizes user's visit conduct in an urban area (see Sect. 1). *(1) Duration:* is the total time of the POI-visits trajectory. It is the time interval, in minutes, between the first and the last POI-visit in a given trajectory. *(2) Nr. POIs:* is the total number of POI-visits in a trajectory and shows the user's willingness to move. *(3) Avg. dwell time:* is the time a user allocates to a specific POI-visit and it is computed by dividing the total duration of a POI-visits trajectory by the number of POI-visits it contains. *(4) Top-[n]:* is the proportion of "must-see" attractions in a user's POI-visits trajectory. In a POI-visits trajectory we count the number of POIs in the trajectory that fall in the top-$n$ ($n = 10, 50, 100$) list of attractions in TripAdvisor "Things to do". Then, we divide that number by the length of the POI-visits trajectory. *(5) Excellence:* is the proportion of fashionable POIs in a user's POI-visits trajectory. Given a POI-visits trajectory, we divide the number of its POIs that have a TripAdvisor's Certificate of Excellence by the trajectory's length.

Clusters of POI-visits trajectories obtained by using the above listed features, are considered as different typologies of users, tight together by similar POI-visits behaviour. For instance, a cluster may consist of tourists that are not particularly knowledgeable about the destination and prefer to visit "must-see" POIs.

### 3.2 User Behaviour Learning

We model the next-POI selection problem as a Markov Decision Process (MDP). A MDP is a tuple $(S, A, T, r, \gamma)$. $S$ denotes the set states, where a state $s$ represents the visit to a specific POI and its context (e.g., the weather condition at the visit time). $A$ is the actions set; an action $a$ models the movement from the previous POI to the target visit POI. $T$ is a set of transition probabilities, where $T(s'|s,a)$ is the probability to move from a state $s$ to a next state $s'$, by performing the action $a$. We also denote a user POI-visits trajectory with $\zeta \in Z$. For instance, $\zeta_{u_1} = (s_2, s_7, s_9)$ represent the user $u_1$ trajectory starting from state $s_2$, passing to $s_7$ and ending in $s_9$. The set of all the observed users' trajectories $Z$ is used to estimate the probabilities $T(s'|s,a)$.

Given a MDP, the goal is to find a policy $\pi^* : S \rightarrow A$ that maximises the cumulative reward $r$ that a decision maker obtains by acting according to $\pi^*$ (optimal policy). The value of taking a specific action $a$ in state $s$ under the policy $\pi$, is computed as $Q_\pi(s,a) = \mathbf{E}^{s,a,\pi}[\sum_{k=0}^{\infty} \gamma^k r(s_k)]$ ($\gamma$ is a discount factor). This is the expected discounted cumulative reward obtained from $a$ in state $s$ and then following the policy $\pi$.

Since (typically) users of a RS scarcely provide feedback on the consumed items (visited POIs) the reward they obtain by consuming an item is rarely known. Hence, we solve the MDP via Inverse Reinforcement Learning [19] which allows to estimate a reward function whose optimal policy (the learning objective) produces actions close to the demonstrated behavior (the user's trajectory). We assume that the reward function $r$ for a state $s$ as $r(s) = \phi(s) \cdot \theta$, is a linear combination of the state's feature vector $\phi(s)$ and the user utility vector $\theta$, which models the unknown user preferences for the various state features. We use Maximum likelihood IRL for learning the target reward function and optimal policy [3].

### 3.3 Clustering Similarly Behaving Users

In order to cluster users with similar visit behaviour, and tailor the recommendations for each cluster, we use a representation of the POI-visit trajectories that contains only the 5 visit behavioural features mentioned in the previous section. For each city we build a matrix $M$ with $|Z|$ rows and 5 columns. Each row represents a POI-visit trajectory $\zeta$ and each column represents a "behavioural" feature. We perform clustering on the standardized (z-score) matrix $M$ by employing the k-Means algorithm.

The optimal number of clusters is found by optimising recommendation precision as discussed in the next section. For the Florence and Rome data sets we found that the optimal numbers of clusters is 6 and 11 respectively. This difference is surely due to the larger number of trajectories and the higher variability of the features values in the Rome data set.

The polar plots in Fig. 1 show how much each "behavioural" feature scores in some of the clusters in the Florence and Rome data sets. For instance, a high "Duration" means that a cluster contains mostly trajectories where the user spent almost the whole day for the visits. Low values for "Excellence" and "Top-n" indicate that the clustered trajectories contain few visits to popular or fashionable places. Overall the clusters in one city show different combinations of feature values. For instance, cluster "Florence 1" has a high number of POIs ("Nr. POIs") that have been visited for a short time (low "Avg. dwell time"), whereas cluster “Florence 2” contains trajectories whose visits last almost as those in “Florence 1” (“Duration”) but that contain lower numbers of POIs whose visit time (on average) is longer (higher “Avg. dwell time”). Interestingly, if we compare the clusters of the two cities we can spot some similarities. For instance, cluster “Florence 1” looks similar to cluster “Rome 3”. To a less extent we can spot similarities in the other clusters like “Florence 2” and “Rome 2” and “Florence 3” and “Rome 1”.

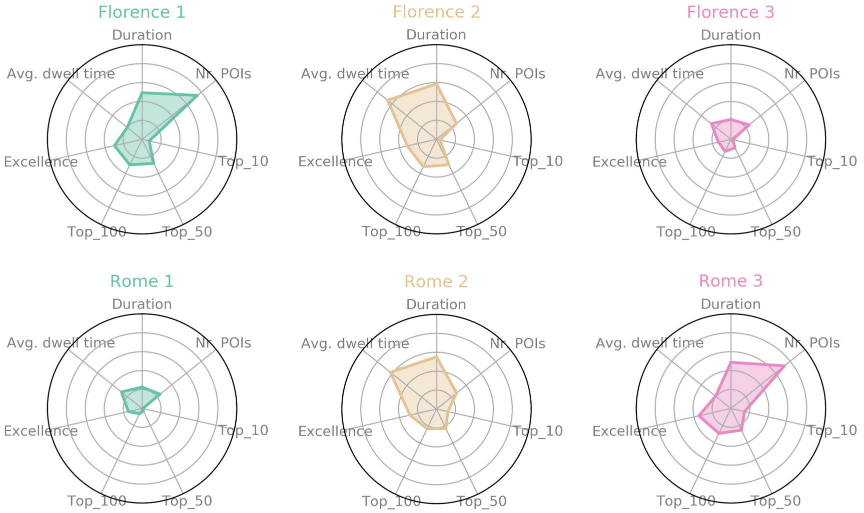

**Fig. 1.** Visit behaviour description in 3 clusters in Florence and Rome (Source: authors)

### 3.4 Recommendation Generation with QBASEX

he IRL-based model here proposed (Q-BASEX) is an extension of Q-BASE [13]. Q-BASE harnesses the behavioural model of the cluster the user belongs to in order to suggest next-POI visit actions the user should make from her current POI-visit (state $s$) [12]. The recommended POI-visit actions $a$ are those with the highest $Q(s,\cdot)$ value in the user current state. However, when users' observations are limited not all the possible contextual situations in a POI and next POI-visit actions combinations may have been observed in the training set. Hence, Q-BASE often is not able to generate a full set of top-n recommendations.

Therefore, we propose here, for a state $s$ for which Q-BASE is not able to generate the required $n$ recommendations, to ignore the information given by the current context of the user in the state $s$, and identify the set of states $gen(s)$ that represent a visit to the same POI of state $s$, but possibly in different contexts. Then, the next POI-visit actions $a$ for which we are able to compute $Q(s',a)$, for states $s' \in gen(s)$, are sorted by $AVG_{s' \in gen(s)}\{Q(s',a)\}$, and the top scoring actions are recommended. We call this new IRL-based RS Q-BASEX (conteXt relaXed). In case a full set of recommendations can not be generated even by ignoring the current user context, Q-BASEX generates recommendations by considering the $s$ predecessor state (if any), hence computing next visit recommendations suited for the previous location of the user; the "previous" state is typically related to the "current" state of the user.

An additional property of Q-BASEX is the capability to generate recommendations for new unseen POIs, i.e., new venues that have not been visited yet by any user, and therefore are not in the training set. Let $\phi(a)$ be the feature vector of $a$, i.e., it is a binary vector containing the same content features modeling a POI in the state model, but here they model the action to move to a POI. Let $a_n \in A_n$ be a new POI-visit action that has not been previously observed (not in the train set). Given the user's current state $s$, by considering the actions for which we are able to compute the value $Q(s, \cdot)$, we compute the (Jaccard index) similarity $sim(\phi(a), \phi(a_n))$ between the POI feature vectors associated to an observed (known) visit action $a \in A_k$ and to the unseen new POI associated to $a_n$. In order to generate next visit recommendations for new POIs using Q-BASEX we compute:

$$Q(s, a_n) = \frac{1}{|A_k|} \sum_{a \in A_k} sim(\phi(a), \phi(a_n)) Q(s, a)$$

The actions that maximise this score are recommended.

# 4 Experimental Study

Our first hypothesis is that by clustering users on the base of behavioural features Q-BASEX can generate better recommendations than a nearest neighbour baseline (SKNN). The second hypothesis is that by assigning a test trajectory to a cluster on the base of the behavioural features Q-BASEX has a better performance than if test trajectories are assigned to a cluster according to content features.

## 4.1 Experimental Strategy

In order to validate the research hypotheses each cluster is partitioned in a train and test set, counting 80% and 20% of the cluster's trajectories respectively. The cluster specific behavioural model is learnt on the train set, whereas the next-POI recommendations (top-1 and top-3) are generated on the POI-visits trajectories in the test set. Actually, each trajectory in the test set is partitioned in two segments: the initial 70% is used for the recommendation generation (it represents the visits performed by the user up to a certain point, reaching the last visited POI) and the remaining 30% is used for the recommendation evaluation (the next POI visits considered to be good for the user). The result show in the next section are the average values of a 3-fold cross-validation evaluation procedure.

To assign a partial user's POI-visit trajectory (test trajectory) to a cluster and compute recommendations we compare two options: a) by using the 5 identified "behavioural" features and, b) by using the contained POIs descriptions (content). In the first assignment a test trajectory is assigned to the closest cluster by computing the euclidean distance between the low-level behaviour representation of the trajectory and the centroids of the clusters. To implement the second assignment, we first build a document-like representation of the POI-visit trajectory by performing the union of the descriptive features of the POIs it contains. In particular, we create a trajectory vector and each entry of the vector counts how many times the corresponding content feature is present in the POIs that fall in the POI-visit trajectory. Then, the vector is normalized to unary length ($L^1$-normalization). The centroids of the clusters' (determined by using the behavioural features) are then computed as average of the trajectory vectors of the contained trajectories. Finally, a test POI-visit trajectory is assigned to the cluster with the smallest cosine distance (from its centroid).

### 4.2 Baseline Recommendation Techniques

We compare the performance of `Q-BASEX` with `SKNN` [10], which is considered to be a strong state of the art next-item recommendation method [11]. It has shown a better accuracy than another IRL-based model presented in [13].

`SKNN` recommends the next-item (visit action) to a user by considering her current session (trajectory) and searching for similar sessions (neighbourhood) in the data-set. These are obtained by computing the binary cosine similarity $c(\zeta, \zeta_i)$ between the current trajectory $\zeta$ and those in the dataset $\zeta_i$. Given a set of nearest neighbours $N_\zeta$, then the score of a visit action $a$ can be computed as:

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

where $\mathbb{1}_{\zeta_n}$ is the indicator function of the trajectory $\zeta_n$: it is 1 if the POI selected by action $a$ appears in the neighbour trajectory $\zeta_n$, and 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: 1785 trajectories for Florence and 3689 for Rome. The actions recommended by `SKNN` are those with the highest scores.

### 4.3 Performance Metrics

Let be $U$ the set of all the users that receive recommendations, $R_{u,s}$ is the recommendation set for the user $u$ in state $s$ (top-1 and top-3), and $Y_{u,s}$ is the the test set of user $u$, that is, the next POI-visit actions observed after state $s$.

**Precision.** is the classical accuracy metric of RSs. Let $\mathbb{1}_{Y_{u,s}}(a)$ be the indicator function of the set $Y_{u,s}$, which is 1 if $a \in Y_{u,s}$ and 0 otherwise. The precision of a recommendation set is: $precision(R_{u,s}) = \frac{1}{|R_{u,s}|} \sum_{a \in R_{u,s}} \mathbb{1}_{Y_{u,s}}(a)$.

**Reward.** measures the increase in reward that the recommended actions give, compared to the next action observed in the user's test set $a_o$. Reward measures the user's gain if she acts according to what is recommended rather than what she is going to do autonomously: $reward(R_{u,s}, a_o) = \frac{1}{|R_{u,s}|}\sum_{a\in R_{u,s}} Q(s,a) - Q(s,a_o)$.

**Similarity.** measures how much the features of the recommended POIs match those of the POI-visit actions in the test set. Let $\phi(a)$ be the feature vector representation of the POI visited by action $a$, and $sim(\cdot,\cdot)$ the Jaccard index similarity. We have: $similarity(R_{u,s}, Y_{u,s}) = \frac{1}{|R_{u,s}||Y_{u,s}|}\sum_{a\in R_{u,s}}\sum_{o\in Y_{u,s}} sim(\phi(a), \phi(o))$.

**I-Coverage.** is the percentage of the relevant items that are actually recommended and ranges in $[0,1]$: $icoverage = \frac{\left|\bigcup_{u\in U} R_{u,s}\cap Y_{u,s}\right|}{\left|\bigcup_{u\in U} Y_{u,s}\right|}$

**Unique.** measures the capability of a RS to suggest diverse items, among the various users, it ranges in $[0,1]$ and it is: $unique = \frac{\left|\bigcup_{u\in U} R_{u,s}\right|}{|U|n}$.

**Popularity.** Let $D_{top50}$ be the set of the top-50 visited POIs in the train set of a RS model and $\mathbb{1}_{D_{top50}}(a)$ its indicator function (it is 1 if $a \in D_{top50}$ and 0 otherwise). We have: $popularity(R_{u,s}) = \frac{1}{|R_{u,s}|}\sum_{a\in R_{u,s}} \mathbb{1}_{D_{top50}}(a)$. We assume that a high popularity POI is likely to be known by the users (e.g., a "must-see" POI) and therefore is not novel, hence novel POIs have low popularity.

## 5 Experimental Results

We first compare the proposed model, `Q-BASEX`, with the `SKNN` baseline. We perform a two-tailed paired t-test with significance level of 0.05 in order to assess if there is a significant difference between the best performing model and the other. If a model is significantly better than the other on a specific metric we underscore in the following tables its performance value. The performance of the two compared RSs when behavioural clustering is employed in the Florence data set is reported in Table 2. `Q-BASEX` outperforms `SKNN` in all the evaluated metrics for top-1 and top-3 recommendations. In particular, `Q-BASEX` recommends next-POI visits that are: more precise (high *Prec*), increase the user's utility (high `Rew`) and closer to the user's expected experience (high *Sim*). Moreover, `Q-BASEX` is less prone to recommend popular places (lower *Pop*) and diversifies the POI-visit suggestions among the users (higher *Unique* and *I-Cov*).

**Table 2.** Recommendation performance on the Florence dataset (Source: authors)

| Model | Top-n | Prec | Rew | Sim | I-Cov | Unique | Pop |
|---|---|---|---|---|---|---|---|
| `Q-BASEX` | 1 | 0,10 | 0,44 | 0,10 | 0,33 | 0,36 | 0,79 |
| `SKNN` | 1 | 0,02 | −0,03 | 0,06 | 0,28 | 0,30 | 0,88 |
| `Q-BASEX` | 3 | 0,09 | 0,26 | 0,10 | 0,53 | 0,21 | 0,78 |
| `SKNN` | 3 | 0,06 | 0,00 | 0,08 | 0,38 | 0,14 | 0,92 |

The RSs performance in the Rome data set is shown in Table 3. Similar observations done for the Florence data set (Table 2) can be made here. The excellent performance of Q-BASEX is confirmed here for all the metrics (both for top-1 and top-3 recommendations). SKNN suggests lower accurate next-POI visits (Prec) that have also lower reward (Rew), compared to Q-BASEX. By looking at the metrics *Sim*, *I-Cov*, *Unique* and *Pop*, we can state that Q-BASEX suggests less popular (low *Pop*) next-POI visits that are also more diverse (high *Unique*) and relevant (low *I-cov* and *Sim*).

**Table 3.** Recommendation performance on the Rome dataset (Source: authors)

| Model | Top-n | Prec | Rew | Sim | I-Cov | Unique | Pop |
|---|---|---|---|---|---|---|---|
| Q-BASEX | 1 | 0,12 | 0,52 | 0,17 | 0,32 | 0,35 | 0,70 |
| SKNN | 1 | 0,00 | −0,07 | 0,12 | 0,30 | 0,34 | 0,79 |
| Q-BASEX | 3 | 0,10 | 0,34 | 0,17 | 0,60 | 0,23 | 0,69 |
| SKNN | 3 | 0,06 | 0,00 | 0,16 | 0,43 | 0,17 | 0,86 |

In summary, Q-BASEX can better support a visitor in identifying POI visits that are relevant and aligned to the users' expected experience (high *Prec*, *Sim* and *Rew*) as well as interesting and diverse (high *I-Cov*, *Unique* and low *Pop*).

Now we focus on the second hypothesis and we show (Table 4) how the performance metrics of Q-BASEX (top-3 recommendations) change if, instead of assign a test user's trajectory to the cluster of similarly behaving users' trajectories, as we have done in the previous experiment, we assign it to a cluster that contains trajectories with similar content features.

**Table 4.** Comparison of the recommendation metrics (top-3) when users are assigned to a cluster on the base of *behavioural* vs. *content* features (Source: authors)

| City | Features | Prec | Rew | Sim | I-Cov | Unique | Pop |
|---|---|---|---|---|---|---|---|
| Florence | Behaviour | 0,09 | 0,26 | 0,10 | 0,53 | 0,21 | 0,78 |
|  | Content | 0,08 | 0,04 | 0,10 | 0,54 | 0,23 | 0,79 |
| Rome | Behaviour | 0,10 | 0,34 | 0,17 | 0,60 | 0,23 | 0,69 |
|  | Content | 0,09 | 0,04 | 0,17 | 0,60 | 0,26 | 0,71 |

In both data sets, clustering by using the behavioural features produces a higher precision, reward, novelty (low *Pop*) and a slightly (non significant) lower diversity (*Unique*). Hence, we can state that the proposed behavioural features (Sect. 3.1) are more suitable than content features to identify similar trajectories to exploit in IRL methods to generate POI visit recommendations.

In conclusion, we have shown that Q-BASEX can better accomplish the task of a RSs in the tourism domain, by suggesting items that are relevant for a user, i.e., with high precision, reward and expected POI-visit experience (similarity), as well as by being able to suggest to the whole user base different items that are also novel.

## 6 Conclusion and Future Works

We have proposed a novel next-POI RS called Q-BASEX that is based on two computational steps: (1) clustering users' trajectories so that each cluster contains users visits' trajectories showing similar behaviour; (2) harnessing a behaviour model, learned for each cluster, to recommend to a user for which a partial POI-visit trajectory is known, next-POI visit actions.

We operationalized a theory-driven schema [2] by identifying visit behaviour features that identifies clusters on the base of *what* and *how* tourists visit a destination. We tested the proposed approach with two datasets consisting of users' POI-visits trajectories in Rome and Florence (Italy), across several dimensions of the recommendation quality and compared its performance to the next-item recommendation model SKNN. Our conclusion is that Q-BASEX can generate recommendations that better match the user's context and interests, and also offers the best combination of precision and novelty, while making suggestions that are more rewarding for the user. Moreover, Q-BASEX effectiveness depends significantly on the used clusters of similarly behaving POI-visit trajectories: how the users visit POIs seems even more important than *what* the users visit.

We plan to conduct a live user-study to assess the users' perceived performance of Q-BASEX and to analyse its fairness in supporting the different users that falls within a cluster, i.e., users with different profiles but treated by Q-BASEX in the same way.

## References

1. Adomavicius G, Tuzhilin A (2011) Context-aware recommender systems. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender systems handbook. Springer, Boston, pp 217–253. https://doi.org/10.1007/978-0-387-85820-3_7
2. Ashworth G, Page SJ (2011) Urban tourism research: recent progress and current paradoxes. Tour Manag 32:1–15
3. Babes-Vroman M, Marivate V, Subramanian K, Littman M (2011) Apprenticeship learning about multiple intentions. In: Proceedings of the 28th international conference on machine learning - ICML'11. pp 897–904
4. Ball P (2010) Wisdom of the fool's choice. Nature
5. van den Berg L, van der Borg J, van der Meer J (1995) Urban tourism: performance and strategies in eight European cities. Avebury, Beatty
6. Brady JV (2003) System and method for locating and notifying a user of a person, place or thing having attributes matching the user's stated preferences, US patent nr. US7071842B1

7. Dacrema Ferrari M, Cremonesi P, Jannach D (2019) Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp 101–109. RecSys '19, Association for Computing Machinery, New York
8. García B (2004) Urban regeneration, arts programming and major events Glasgow 1990, Sydney 2000 and Barcelona 2004. Int J Cult Policy 10(1):103–118
9. Höpken W, Müller M, Fuchs M, Lexhagen M (2020) Flickr data for analysing tourists' spatial behaviour and movement patterns: a comparison of clustering techniques. J Hosp Tour Technol 11(1):69–82
10. Jannach D, Lerche L (2017) Leveraging multi-dimensional user models for personalized next-track music recommendation. In: Proceedings of the symposium on applied computing - SAC'17, pp 1635–1642
11. Ludewig M, Jannach D (2018) Evaluation of session-based recommendation algorithms. User Model User-Adapt Interact 28(4–5):331–390
12. Massimo D, Ricci F (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, pp 402–406
13. Massimo D, Ricci F (2020) Enhancing travel experience leveraging on-line and off-line users' behaviour data. In: IUI '20: 25th international conference on intelligent user interfaces, Cagliari, Italy, March 17–20, 2020, Companion, pp 65–66. ACM
14. McNee S, Riedl J, Konstan J (2006) Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI '06 extended abstracts, pp 1097–1101
15. Mobasher B, Dao H, Luo T, Nakagawa M (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, pp 669–672
16. Modsching M, Kramer R, Hagen KT, Gretzel U (2008) Using location-based tracking data to analyze the movements of city tourists. Inf Technol Tour 10:31–42
17. Moling O, Baltrunas L, Ricci F (2012) Optimal radio channel recommendations with explicit and implicit feedback. In: Proceedings of the 6th ACM conference on recommender systems - RecSys'12, p 75
18. Muntean CI, Nardini FM, Silvestri F, Baraglia R (2015) On learning prediction models for tourists paths. ACM Trans Intell Syst Technol 7(1):1–34
19. Ng A, Russell S (2000) Algorithms for inverse reinforcement learning. In: Proceedings of the 17th international conference on machine learning - ICML '00, pp 663–670
20. Shani G, Heckerman D, Brafman RI (2005) An MDP-based recommender system. J Mach Learn Res 6:1265–1295
21. Yao D, Zhang C, Zhu Z, Huang J, Bi J (2017) Trajectory clustering via deep representation learning. In: 2017 international joint conference on neural networks (IJCNN), pp 3880–3887
22. Zheng Y (2015) Trajectory data mining: an overview. ACM TIST 6:29:1–29:41