Tangible Decision-making in Sensors Augmented Spaces*

David Massimo and Francesco Ricci

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

Abstract. Recommender Systems (RSs) are web tools aimed at easing users' online decision-making. Here we propose a complementary scenario: supporting (tangible) decision-making in the physical space. In particular, we propose a novel RS technology that harness data coming from a sensor augmented environment, e.g., a Smart City. In such setting, users' movements can be tracked and the knowledge of their choices (visit to points of interest, POIs) can be used to generate recommendations for not yet visited POIs. The proposed technique overcome the inability of current RSs to generalise the preferences directly derived from the user's observed behaviour by decoupling the learning of the user behaviour (predicted choices) from the recommender model (recommended choices). In our approach we apply clustering to users' observed sequences of choices (i.e., POI visit trajectories) in order to identify like-behaving users and then we learn the optimal user behaviour model for each cluster. Then, by harnessing the learned optimal behaviour model we generate novel and relevant recommendations, which provide useful information in addition to choices that the user will make without any recommendation (predicted choices). In this paper we summarise the proposed RS technology; we describe its performance across different dimensions in an offline test and a users study by comparing the proposed technique with session-aware nearest neighbour based baselines (SKNN). The offline analysis results show that our approach suggests items that are novel and increases the user's satisfaction (high reward), whereas the SKNN approaches are good at predicting the exact user behaviour. Interestingly, the online results show that the proposed approach excels in what a (tourism) RS should do: suggesting items that the user is unaware of and also relevant.

Keywords: Recommender Systems · Inverse Reinforcement Learning · Clustering · User Study.

1 Introduction

Finding relevant information in an online catalogue is not an easy task. Users may be exposed to a large variety of content, incurring in information overload

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

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2495/paper7.pdf

[15], and therefore may make poorly informed decisions. In order to ease human decision-making Recommender Systems (RSs) have been proposed. A RS is a web-tool that identifies for a user items that are (potentially) appropriate for her current need. Since users' preferences and behaviour may also be influenced by contextual factors, such as, the weather conditions at the time of the item consumption, context-aware RSs have been introduced [1]. Moreover, in order to leverage the knowledge derived from the order in which users consume items, pattern-discovery [10, 4, 14] and reinforcement learning [16, 11] approaches have also been proposed. The first approach extracts common patterns from users' behaviour logs and learns a predictive model of the next user action. The latter generates recommendations by using the optimal choice model (policy) of the user. In both models the recommendation generation process is strictly tight to the learnt user's behaviour, i.e., they suggest the user's predicted next choice. Moreover, the first approach can only suggest items that have been already observed, while the second assumes that the utility the user gets from her choices is known in advance. This is contrasting with the tendency of users to rarely provide an explicit feedback (e.g., ratings).

RSs technology has been mostly applied to the web scenario, where users interact with online content. With the advent of sensor augmented spaces, like Smart Cities, where sensors collects and leverages data to handle assets and resources efficiently, RS technology could be applied to ease users' (tangible) decision-making while they interact with the physical space. A RS can leverage the observations of users' choices recorded by the sensors. In fact, our application domain is tourism, where a user acts in different contexts and performs decisions about what to visit in a sequential fashion. E.g., a tourist decides which point of interest (POI) she would like to visit next, given her past visit choices and contextual conditions. In a sensor augmented space the tourist's sequences of choices (i.e., trajectory) may be recorded by sensors and can be leveraged to identify relevant and useful next-POI visits for tourist.

We propose a novel RS context-aware technique that, not only eases the (tangible) decision-making of users while they interact with a sensor augmented space, but also overcomes the main problem of the aforementioned RS solutions: the inability to generalize from the observed data and, consequently, the poor novelty of the recommendations. Hence, we have devised a RS model that can explain and generalize from observed behaviours in order to generate non-trivial and relevant recommendations for a user.

Our RS approach models with a reward function the "satisfaction" that a POI, with some given features, gives to a user. The reward function is learnt by using the observation of the users' sequences of visited POIs and is estimated by Inverse Reinforcement Learning (IRL) [13], a behaviour learning approach that is widely used in automation and behavioural economics [5, 3, 18]. Moreover, since it is difficult to have at disposal the knowledge of a consistent part of a new visitor's travel related choices, which would be needed to learn the reward function of a single individual, IRL is instead applied to clusters of users, and the 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.

In this paper we show the two main component of the proposed RS technology: clustering of users in different tourist types in order to learn generalized user behaviour models via IRL; recommendation strategies that harness the learnt behaviour models in order to generate novel and relevant suggestions for the user. Moreover, we discuss the performance of the proposed method across several dimensions in an offline study. The results indicate that the proposed IRL-based solution excels in suggesting novel and rewarding items, whereas a (SKNN-based) pattern-discovery baseline has a higher precision. We conjecture that the lack of precision of the proposed solution is due to the fact that SKNN-based methods favour items that appears frequently in the data, i.e., items that are popular. To further study this aspect we hybridize the proposed RS technique with an item popularity scoring technique and show that our conjecture holds: biasing the IRL-based model with item popularity allows the model to practically equals the precision of the KNN-based baseline.

The remainder of the paper is structured as follows. In Section 2 we describe the formalisation of the recommendation problem, how users are clustered in tourist types and how the user's action-selection policy (i.e., behaviour) is learned via Inverse Reinforcement Learning. Then we detail two recommendation strategies: the strategies presented in [8, 9] and an additional model that combines the proposed IRL-based approach with item popularity. In section 3 we present the baselines, the metrics and the evaluation procedure of the offline study and the user study. In Section 4 we report the experimental results. Finally, in Section 5 we state the conclusion.

2 Method

2.1 User Behaviour Modelling

User (tourist) behaviour modelling is here 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. A user that is situated in a specific POI and context can reach all the other POIs in a new context. T is a finite set of probabilities $T(s'|s, a)$: the probability to make a transition from state s to s' when action a is performed. For example, a user that visits Museo di San Marco in a sunny morning (state s_1) and wants to visit Palazzo Pitti (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) with transition probabilities $T(s_2, a_1|s_1) = 0.2$ and $T(s_3, a_1|s1) = 0.8$. 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 visited a POI and not another (nearby) one is because she believes 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.

2.2 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 [17].

We denote with ζ_u a user u trajectory, which is a temporally ordered list of states (POI-visits). For instance, $\zeta_{u_1} = (s_{10}, s_5, s_{15})$ represents 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 (i.e., visited POIs), the reward a user gets by consuming an item is not known. Therefore, the MDP cannot be solved by using standard Reinforcement Learning techniques. Instead, by having at disposal only a set of POI-visit observations of a user (i.e., the users' trajectories), a MDP could be solved via Inverse Reinforcement Learning (IRL) methods [13]. 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].

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) [9, 8]. 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 done by employing Non Negative Matrix Factorization (NMF) [6] on document like representations of the trajectories (features are treated as keywords).

2.3 Recommending Next-POI visits

Here we present the above mentioned IRL-based recommendation techniques: Q-BASE shown in [8] as well as a novel method that hybridizes Q-BASE with

the popularity of an item.

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 PUSH. In order to recommend more popular items, we propose to hybridise the Q-BASE model with the recommended item popularity, i.e., a score proportional to the probability that a user visit the item.

Q-POP PUSH scores the (potential) 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 $pop(a)$, the scaled (i.e., min-max scaling) counts $c_Z(p)$ (in the data set Z) of the occurrences of the POI-visit p selected by the action a (an action corresponds to the visit to of a single point).

3 Evaluation

3.1 Dataset

In this study we used an extended version of the POI-visit data-set presented in [12]. It consists of tourist trajectories reconstructed from the public photo albums of users of the Flickr[1] platform. The trajectory extraction process is as follow, 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 represented 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 (POI historic period, POI type and related public figure). In this paper we use an extended version of the dataset that contains 1668 trajectories and 793 POIs. Trajectories clustering identified 5 different clusters, as in the previous study.

3.2 Baselines

We compare here the performance of the recommendations generated by the proposed IRL-based methods with two nearest neighbor baselines: session-based KNN (SKNN) and sequence-aware SKNN (s-SKNN).

SKNN [4] seeks for similar users in the system stored logs (trajectories) and identifies the next-item (POI) to be recommended, given the current user log (user trajectory), by using a classical collaborative filtering scoring rule.

s-SKNN[7] uses again the classical collaborative filtering rule but weights the neighbours importance by weighting more those containing the most recent

[1] www.flickr.com

items (recent POIs in the user trajectory). These methods have been applied to different next-item recommendations tasks showing good performance.

3.3 Metrics

The proposed recommendation strategies were benchmarked by using several metrics. The *reward* metric measures the average increase of reward of the recommended actions compared to the observed one (in the test part of the trajectory). It is the aggregated difference of the recommended POI-visits Q values and the Q value of the observed (test) visit. *Dissimilarity* measures how much the recommendations are dissimilar from the observed visit and ranges in $[0, 1]$. *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. Detailed definitions of these metrics can be found in [8]. *Precision* is the percentage of recommended visits that match the observed one, hence it shows to what extent the system suggests the actions actually performed by the user.

3.4 Offline Study

Initially, for each cluster, 80% of the trajectories were assigned to the train set and the remaining 20% to the test set. Then, for each cluster, the train set data was used to learn the generalised user behaviour model for that cluster. Afterwards, in order to compute and evaluate recommendations, the trajectories in the test set were split in two parts: the initial 70% of each trajectory was considered as observed by the system and used to generate next action recommendations, while the remaining part (30%) was actually used as test part in order to asses the evaluation metrics. The SKNN-based baselines do not use clustering, hence they were trained on all the trajectories in the train set and the test set trajectories were split in observed and test parts as before. All the models hyper-parameters have been selected via 5-fold cross validation.

3.5 Online Study

We also conducted an observational study with real users. They interacted with an online system that we developed to assesses the novelty and user satisfaction for the recommendations generated by the Q-BASE model, the Q-POP PUSH model and the same SKNN baseline used in the offline study. In the online system the users can enter the set of POI that they have already visited in the city of Florence and can receive suggestions for next POIs to visit. In particular, the user can mark the suggestions with the labels “visited”, “novel”, “liked”. To avoid biases in the recommendation evaluation we do not reveal to the user which recommendation algorithm produces which POI recommendation. The suggestions that the user evaluates is a list that aggregates the top-3 suggestions of each algorithm without giving to any algorithm a particular priority.

4 Results

4.1 Offline Study Results

The compared recommenders' performance for top-1 and top-5 next-POI visit recommendations are shown in Table 1. One can observe that Q-BASE allows users to obtain larger reward, compared to SKNN and s-SKNN. While, as expected, SKNN-based baselines have the best precision, as they tends to suggest next-POIs that the user would anyway visit. Interestingly, SKNN and s-SKNN perform very similarly. Hence, in this data-set, the sequence-aware extension of SKNN does not provide any performance improvement. These results confirm a previous analysis [8, 9].

By looking at the performance of Q-POP PUSH we see that a stronger popularity bias enables the algorithm to generate recommendations that are more precise. In fact, the precision of Q-POP PUSH is equal to that of SKNN and s-SKNN. But, as expected, reward and novelty are penalised.

Table 1. Recommendation performance

Models	**Q-BASE**	**Q-POP**	**PUSH SKNN**	**s-SKNN**
Rew@1	0.073	-0.002	-0.007	-0.009
Prec@1	0.043	0.099	0.109	0.109
Nov@1	0.061	0.000	0.000	0.000
Rew@5	0.032	-0.009	-0.010	-0.010
Prec@5	0.045	0.060	0.068	0.063
Nov@5	0.122	0.000	0.000	0.000

4.2 Online Study Results

The results of the online study are shown in Table 2. This table shows the probabilities that a user marks as "visited", "novel", "liked" 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. 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.

We note that the POIs recommended by SKNN and Q-POP PUSH have the highest probability (24%) that the user have already visited them, and the lowest probability to be considered as novel. Conversely, 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.

Table 2. 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 & Novel	0.091	0.076	0.082

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 argue that this is the primary goal of a recommender system: to enable users to discover novel items that are interesting for them. 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%).

5 Conclusion

In this paper we presented a new next-POI RS technique that harness a generalised tourists behaviour model. The tourist behaviour model is learnt by firstly clustering users' POI-visit trajectories and then by solving an Inverse Reinforcement Learning problem which determines, for each cluster, the reward function and the optimal POI selection policy. The proposed recommendation strategies (Q-BASE and Q-POP PUSH) adapt the next visit-action recommendations to the learned model. We show with an offline experiment that the proposed Q-BASE model maximises the reward the user gains while discovering relevant, novel and non-popular POIs. Moreover, the two SKNN-based baselines shows a better offline accuracy. We hypothesised that users like more the recommendations produced by Q-BASE and that the poor offline accuracy of these models, compared to SKNN-based approaches, is due to the absence of a popularity bias in the recommendation generation. Therefore, we hybridize Q-BASE with POI popularity and show that the hybrid model (Q-POP PUSH) substantially equals the SKNN baselines. With an online test we show that the Q-BASE model excels in suggesting novel items that are also liked ("liked and novel") by the users.

We plan to extend the presented analysis by conducting an evaluation with tourists interacting with real systems while on the move[2].

[2] www.wondervalley.unibz.it and https://beacon.bz.it/wp-6/beaconrecommender/

References

1. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 217–253 (2011)
2. Babes-Vroman, M., Marivate, V., Subramanian, K., Littman, M.: Apprenticeship learning about multiple intentions. In: Proceedings of the 28th International Conference on Machine Learning - ICML'11. pp. 897–904 (2011)
3. Ermon, S., Xue, Y., Toth, R., Dilkina, B., Bernstein, R., Damoulas, T., Clark, P., DeGloria, S., Mude, A., Barrett, C., Gomes, C.P.: Learning Large Scale Dynamic Discrete Choice Models of Spatio-Temporal Preferences with Application to Migratory Pastoralism in East Africa. pp. 644–650 (2015)
4. Jannach, D., Lerche, L.: Leveraging Multi-Dimensional User Models for Personalized Next-Track Music Recommendation. In: Proceedings of the Symposium on Applied Computing - SAC'17. pp. 1635–1642 (2017)
5. Kennan, J., Walker, J.R.: The Effect of Expected Income on Individual Migration Decisions. Econometrica **79**(1), 211–251 (2011). https://doi.org/10.3982/ECTA4657
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
7. Ludewig, M., Jannach, D.: Evaluation of session-based recommendation algorithms. User Model. User-Adapt. Interact. **28**(4-5), 331–390 (2018)
8. Massimo, D., Ricci, F.: 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 (2018)
9. Massimo, D., Ricci, F.: 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. pp. 3–14 (2019)
10. Mobasher, B., H. Dao, T. Luo, Nakagawa, M.: 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 (2002)
11. Moling, O., Baltrunas, L., Ricci, F.: Optimal radio channel recommendations with explicit and implicit feedback. In: Proceedings of the 6th ACM conference on Recommender systems - RecSys '12. p. 75 (2012)
12. Muntean, C.I., Nardini, F.M., Silvestri, F., Baraglia, R.: On Learning Prediction Models for Tourists Paths. ACM Transactions on Intelligent Systems and Technology **7**(1), 1–34 (2015)
13. Ng, A., Russell, S.: Algorithms for inverse reinforcement learning. In: Proceedings of the 17th International Conference on Machine Learning - ICML '00. pp. 663–670 (2000)
14. Palumbo, E., Rizzo, G., Baralis, E.: Predicting Your Next Stop-over from Location-based Social Network Data with Recurrent Neural Networks. In: RecSys '17, 2nd ACM International Workshop on Recommenders in Tourism (RecTour'17), CEUR Proceedings Vol. 1906. pp. 1–8 (2017)
15. Roetzel, P.G.: Information overload in the information age: a review of the literature from business administration, business psychology, and related disciplines with a bibliometric approach and framework development. Business Research (2018)

16. Shani, G., Heckerman, D., Brafman, R.I.: An mdp-based recommender system. Journal of Machine Learning Research pp. 1265–1295 (2005)
17. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction (Second edition, in progress). The MIT Press (2014)
18. Ziebart, B.D., Maas, A., Bagnell, J.A., Dey, A.K.: Maximum entropy inverse reinforcement learning. In: Proceedings of the 23rd National Conference on Artificial Intelligence - AAAI'08. pp. 1433–1438 (2008)