Harnessing a Generalised User Behaviour Model for Next-POI Recommendation*

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

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

ABSTRACT

Recommender Systems (RSs) are commonly used in web applications to support users in finding items of their interest. In this paper we propose a novel RS approach that supports human decision making by leveraging data acquired in the physical world. We consider a scenario in which users' choices to visit points of interests (POIs) are tracked and used to generate recommendations for not yet visited POIs. We propose a novel approach to user behaviour modelling that is based on Inverse Reinforcement Learning (IRL). Two recommendation strategies based on the proposed behaviour model are also proposed; they generate recommendations that differ from the common approach based on user next action prediction. Our experimental analysis shows that the proposed approach outperforms state of the art models in terms of the overall utility the user gains by following the provided recommendations and the novelty of the recommended items.

KEYWORDS

Recommender Systems; User Modelling; Inverse Reinforcement Learning; Tourism

ACM Reference Format:
David Massimo and Francesco Ricci. 2018. Harnessing a Generalised User Behaviour Model for Next-POI Recommendation. In *Twelfth ACM Conference on Recommender Systems (RecSys '18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3240323.3240392

1 INTRODUCTION AND STATE OF THE ART

Recommender Systems (RSs) are software tools that support user decision making by identifying personalised and relevant items [16]. Since users' preferences and choices are affected by contextual factors (e.g., being in a group or the items consumed before) context-aware RSs (CARSs) have been introduced [1]. Moreover, in order to leverage the knowledge of the specific order in which the user consumes items, pattern-discovery [5, 9, 14] and reinforcement learning [10, 17] have also been proposed. In the first approach, common patterns in users' behaviour logs (users' choices) are identified and a predictive model of the next user choice is learnt. In the second one, recommendations are generated by exploiting an optimal choice model (policy) that is learnt from the utility (reward) the users receive from their choices. A common feature of both approaches is that the recommended items are the predicted next choices of the target user. Moreover, the first approach can only suggest items that have been already observed, i.e., it suffers from the new item problem, whereas the second one assumes that the system knows the utility the user gets from her actions, while in practice users rarely provide explicit feedback (e.g., ratings). One major drawback of these techniques is that they tend to generate recommendations that lack novelty and that therefore tend to be not interesting for the user [19].

In order to overcome the above mentioned problems we propose a recommendation technique that harnesses a generalised user behavioural model that is built by observing a group of users (cluster) "similar" to the target one. Moreover, we decouple user behaviour learning from recommendation generation. We propose this approach as a general solution for generating non-trivial recommendations that focus on the suggestion of novel and relevant items for the target user. We show that the proposed approach provides recommendations that, in our application scenario, allow tourists to maximise their estimated reward (utility) by visiting the suggested POIs, rather than those actually visited (which are normally used as ground truth for testing a RS). Here we assume that the reward a user obtains by visiting a sequence of POIs, which is called a "trajectory", is the sum of the rewards for the POIs belonging to the trajectory and it is estimated by learning the reward function of the similar users group.

The paper is structured as follows. Firstly, we describe our original formalisation of the recommendation problem and the dataset that we have used to test our approach. Then, the two main components of the user behaviour model are presented: the clustering of users in tourist types and the user's action-selection policy learned via Inverse Reinforcement Learning. Afterwards, we detail two recommendation strategies, the compared baseline recommendation methods, the considered evaluation metrics, the experiment procedure and finally the empirical results.

2 PROPOSED APPROACH

2.1 Dataset

We have assessed the validity of the proposed recommendation approach on a dataset consisting of 575 geo-localized temporally ordered trajectories of users' POI-visits, recorded via GPS sensors

*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.

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240392

in the historic centre of Florence (Italy)[11]. We added to the POI-visits data describing the POIs itself and the context of the visits. In fact, we collected an hourly weather summary (e.g., cloudy), temperature (e.g., cold) and daytime (e.g., evening) for each POI-visit in a trajectory by querying a weather service[1]. Moreover, we identified for each POI its category (e.g., monument), the historical period (i.e., century) and one historical person related to the POI. In the 204 POIs appearing in the trajectories we identified 12 different POI categories, 17 historical periods and 76 historical persons.

2.2 Problem Modeling

Markov Decision Process Model. We model the tourist trajectory generation task as a finite Markov Decision Process (MDP). A MDP is defined by a tuple (S, A, T, r, γ). S is a finite set of states and in our scenario a state represents the visit to a POI in a specific context (weather, temperature and day). For instance, a tourist that visits Florence can be at the Battistero (POI) in a cloudy, cold morning (context). A is a finite set of actions, which in our application represent the decision to move to a POI (so they match one-to-one to the POIs). T is a finite set of probabilities $T(s'|s, a)$, to make a transition from state s to s' when action a is performed. For example, a user that visits Battistero in a cloudy morning (state s_1) and wants to visit the Uffizi Gallery (action a_1) in the afternoon can arrive to the desired POI with either a cloudy sky (state s_2) or a clear sky (state s_3) with transition probabilities $T(s_2, a_1|s_1) = 0.5$ 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 supposed to be *unknown* and must be learnt, i.e., we take the restrictive assumption that we do not know the utility 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, then she believes that the first POI gives her a larger utility/reward than the second. Finally, $\gamma \in [0, 1]$ is used to discount future rewards with respect to immediate ones.

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. 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].

As we mentioned earlier, in a RS application the reward obtained by a user when visiting a state (i.e., the r function) is usually unknown because users scarcely provide feedback. Therefore, we are interested in determining the reward function r from the bare observations of the decision maker transitions from state to state; this problem is solved by Inverse Reinforcement Learning (IRL).

[1] https://darksky.net

Inverse Reinforcement Learning (IRL). IRL algorithms can reconstruct the reward function r and the optimal action-selection policy π^* of a user u from the set of her observed trajectories. We assume (as in [13]) that r is a linear function, $r(s) = \phi(s) \cdot \theta$, of the state s feature vector $\phi(s)$ and the user utility vector θ, which models the unknown user preference for the state features. IRL algorithms derive the user's action-selection policy from the learned reward function r by assuming that users act in order to maximise the reward.

In our application the vector $\phi(s)$ is binary and represents the presence/absence of the following attributes: weather f_w, where $w \in \{$*clear, partly cloudy, mostly cloudy*$\}$; temperature f_t, where $t \in \{$*cold,warm*$\}$; daytime f_d, where $d \in \{$*morning, afternoon, evening*$\}$; POI category f_c, where $c \in \{$*museum, monument,..., palace*$\}$; historic period f_h, where $h \in \{$*century 3,..., century 20*$\}$; related person f_r, where $r \in \{$*Brunellschi,...,Vasari*$\}$. Finally, we consider a qualitative measure of the distance travelled by the user from the previous state-visit to the current state-visit f_d, where $d \in \{$*close, far*$\}$. In total we have 114 Boolean features ($F = 114$), 105 representing the POI ($P = 105$) and 9 representing the context ($C = 9$). Since actions in A represent POIs and a state vector combines a POI with context features we also define $\phi(a)$ as the feature vector part describing the POI.

We use a specific IRL algorithm called Maximum Log-likelihood (MLIRL) [2]. MLIRL combines many positive features of other IRL models [12, 15, 20]: it assumes a prior knowledge of the user preference vector to estimate an initial reward function that is then adjusted by looking for a maximum likelihood model that can justify observed trajectories; it optimizes user behaviour via a gradient method and assumes that each user randomizes the action selection process at the level of individual choices.

Generalised User Behaviour. In our application scenario, as it often happens, there are not many visits trajectories per user. Actually, we have only one trajectory per user. Therefore, we propose to group users' trajectories with a clustering technique and then to learn a "general" user behavioural model common for all the users/trajectories in a cluster. The additional advantage of this approach is to minimize the impact of suboptimal behaviour that could influence some trajectories. By applying MLIRL on each cluster of trajectories we therefore learn cluster specific reward functions and behaviour models of the users in each cluster. This is the optimal policy that dictates for each state the best action (next POI visit) the users in a cluster should take in order to maximise their reward.

Clustering the trajectories is implemented with Non Negative Matrix Factorization (NMF) [8]. NMF groups trajectories according to a common semantic structure that can explain the resulting clusters. NMF extracts topics, i.e., lists of words, that describe groups of documents. Hence, in order to apply NMF, we built a document-like representation of a user trajectory that is based on its features (terms) associated to its states. By considering all the trajectories (in the set Z) we produced a corpus of documents. We denote with D the *tf-idf* matrix representation of the obtained corpus. Columns in D represents specific terms and rows corresponds to trajectories. The matrix D has size $|Z| \times F$. NMF approximates the matrix D with the product of two non-negative matrices W (of size $F \times K$) and H (of size $|Z| \times K$). The matrix H identifies which topics (columns)

Table 1: Top 10 terms in the five topics extracted from the trajectory data set and number of trajectories assigned to each topic (cluster).

#Term	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	cold	morning	warm	clear	evening
2	afternoon	cold	afternoon	cold	cloudy
3	cloudy	cloudy	cloudy	monument	cold
4	monument	monument	monument	warm	warm
5	century 14	century 16	century 14	century 15	monument
6	century 15	square	century 16	morning	century 16
7	Brunelleschi	Vasari	century 15	afternoon	church
8	century 16	15	palace	century 14	Giambologna
9	square	century 19	building	evening	14
10	Giotto	building	Vasari	Donatello	century 19
#Traj.	270	177	277	114	131

are more relevant for each user trajectory (row), and using it we assigned a trajectory to the topics that in its corresponding row have values larger than 0.5 (similarly to [7]). Hence, each topic defines a cluster of trajectories. Moreover, a topic, can be described by its top terms, i.e., those with the largest values in the corresponding row in W. In Table 1 are reported the top-10 terms per extracted topic (cluster) and the number of trajectories per cluster. Five topics/clusters have been determined by conducting a stability analysis on our data set, as suggested in [4].

Recommendations. We now propose two recommendation strategies that suggest relevant next-visit actions. Given the observed POI-visits trajectory of a user, and the general user behavioural model of the cluster the user belongs to, these strategies suggest viable next POI-visit actions for the user.

Cluster Behaviour Based Recommendations (CBR)**.** The first strategy can be used when few observations, e.g., only the current sequence of actions performed by the user, are available. In this case, the general user behaviour of the cluster the user belongs to is used to suggest the optimal action this user should take after the last visited POI. The optimal action is the action with the highest Q value in the user current state. We note that this strategy can recommend novel visit actions that the group generalised behaviour evaluates as optimal on the base of the features of the suggested POI.

Cluster Behaviour Hybrid Recommendations (CBHR)**.** When more visit actions observations for a user are available, recommendations can be generated by combining the group generalised user behaviour (as it is used in the previous strategy) with a user specific preference model based only on the user observed visits. In order to do that we compute two scores. The first one $R_G(s,a)$ is the normalised $Q(s,a)$ (dividing $Q(s,a)$ by the maximum of Q on A). The second score, $R_u(s,a)$, is obtained as follows. The user u specific preference model is represented by a vector α_u of 105 Boolean features, equal to the number of features describing the POIs. The entries in this vector count the corresponding features that are true in the POIs present in the user u trajectory. Then the vector α_u is normalised so that the sum of its entries is 1; the user specific score is $R_u(s,a) = \alpha_u \cdot \phi(a)$. Finally, recommended visit actions are ranked according to the sum of the scores described earlier: $R(s,a) = R_u(s,a) + R_G(s,a)$. For instance, if a commuter is understood to like arts because she mostly visits museums, and an exhibition is estimated to be an optimal choice for her group, then the system could recommend it to her.

3 EVALUATION

3.1 Baseline Recommendation Methods

We compare our recommendation approach with two baseline methods. The first one is a **kNN** based approach that does not distinguish between user behaviour learning and recommendation generation [5, 6]. kNN takes the currently observed target user trajectory as input and seeks for other user trajectories in the data (train) that contains the same actions found in the target user trajectory. Then, the next-action recommendation is selected among those contained in the most similar trajectories in the (train) data. More specifically, the similarity of trajectories is computed as the cosine similarity of the current target user trajectory and the trajectories in the train data. For the computation of action recommendations, a scoring function is employed: given a target user trajectory ζ, its set of neighbour trajectories N_ζ and the similarity function $c(\zeta_1, \zeta_2)$ of trajectories ζ_1 and ζ_2, the score of a target action a is: $score(a,\zeta) = \sum_{\zeta_n \in N_s} c(\zeta, \zeta_n) 1_{\zeta_n}(a)$. With 1_{ζ_n} we denote the indicator function that evaluates to 1 if the POI selected by action a appears in the neighbour trajectory ζ_n (0 otherwise). The selected actions are those with the highest scores.

The second baseline recommendation method is Adaptive Linear Mapping Model (**ALMM**) [3]. It predicts the next POI visit of a user given her current trajectory by exploiting POIs content features. The model exploits the users' trajectories to build a POI-to-POI tensor matrix, where the user is the third dimension. In order to learn the missing POI-to-POI transition probabilities the algorithm factorizes this tensor by exploiting POI content information. Recommended next visit actions are those with the highest predicted probabilities given the current last visited POI of the user.

3.2 Evaluation Metrics

We have identified a set of evaluation metrics that assess different properties of the RS and can help to better estimate the user's satisfaction for the recommended items. In fact, this cannot be estimated only with the precision to predict the test POIs actually visited by the user. We denote $Rec_{u,s}$ a list of recommendations for the user u in state s, and a_o the observed (next) POI-visit (test item). The binary feature vector describing the POI corresponding to action a is denoted with $\phi(a)$, similarly to the feature vector that represents a state.

Reward. This metric measures the average increase in reward the recommended actions give compared to the observed one: $reward(Rec_{u,s}, a_o) = (\sum_{a \in Rec_{u,s}} Q(s,a) - Q(s,a_o))/|Rec_{u,s}|$.

Dissimilarity. Let $\phi_f(a)$ be the f feature of a (ranging in $[0,1]$); this metric measures how much the recommendations are dissimilar from the observed visit:

$$dissimilarity(Rec_{u,s}, a_o) = \frac{\sum_{a \in Rec_{u,s}} \sum_{f=1}^{P} \frac{|\phi_f(a) - \phi_f(a_o)|}{P}}{|Rec_{u,s}|}$$

Novelty. This metric 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 evaluates to 1 if $a \in U$ and 0 otherwise), the novelty is defined as follows: $novelty(Rec_{u,s}) = \frac{\sum_{a \in Rec_{u,s}} 1_U(a)}{|Rec_{u,s}|}$.

Precision. Let obs_u be the set of observed POI-visit actions in the user u trajectory (test set). The indicator function $1_{obs_u}(a)$ returns 1 if $a \in obs_u$ and 0 otherwise. Precision is computed as follows: $precision(Rec_{u,s}) = (\sum_{a \in Rec_{u,s}} 1_{obs_u}(a))/|Rec_{u,s}|$.

3.3 Experimental Strategy

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 baseline RS methods (kNN and ALMM) do not use clustering, hence they were trained on all the trajectories in the train set (the union of the clusters' train sets) 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.

4 RESULTS

The results of the experimental procedure are shown in Table 2. The evaluation metrics have been computed for the top-1 and top-5 next-POI visit recommendations[2]. For the RS strategies proposed in this paper we show the average of the average results obtained in the clusters. This allows us to better compare them with the baseline approaches.

By comparing the next-POI actions recommendations and the true (next) action of the users (test) in terms of reward, the proposed strategies (CBR and CBHR) generate recommendations that allow users to obtain larger rewards. This suggests that these recommendations can be of large interest for users (as we will show next). The motivation of the poor performance of the baselines can be related to the fact that both models are myopically targeted to predict the next action taken by the user, which may be suboptimal.

In terms of recommendations dissimilarity, interestingly, all the recommendation approaches score equally high. Taken together, i.e., the observed reward and recommendations dissimilarity, one can make some conjectures about the user perceived quality of the recommendations. Namely, the proposed approaches suggest POI visits that are both different from what the user would anyway do, hence they are informative, and maximise users' satisfaction (reward). Conversely, even though baseline recommendation methods lead to comparable results in terms of recommendations dissimilarity, they suggest POI visits that give to the users rewards smaller than what the users obtain from the POIs they actually visited (test set).

Considering now the precision of the recommendation methods, the experimental results confirm what we stated in the Introduction, namely, that kNN tends to recommend the POI visits that the user would make anyway. In fact, in terms of precision, kNN outperforms the other RS approaches.

[2]The differences between the strategies CBR, CBRH and the others are significant (Wilcoxon signed-rank test) with $p < 0.01$ for the reward and novelty metrics.

Table 2: Evaluation results

Models	CBR	CBHR	kNN	ALMM
Reward@1	0.395	0.223	-0.437	-0.874
Dissimilarity@1	0.859	0.841	0.839	0.831
Precision@1	0.086	0.046	0.135	0.046
Novelty@1	0.267	0.271	0.000	0.040
Reward@5	0.319	0.193	-0.809	-0.821
Dissimilarity@5	0.873	0.814	0.851	0.832
Precision@5	0.021	0.043	0.069	0.034
Novelty@5	0.487	0.376	0.000	0.216

Conversely, a significant benefit brought by the proposed clustering approach is the larger novelty of the recommendations, in comparison to the novelty of the baseline methods. In fact, both CBR and CBHR recommend items that are estimated to be appropriate for the users according to their features, hence, they can generalise the recommendations by considering as appropriate POIs that possess the features of the POIs visited by the user. This, ultimately, make possible for these methods to recommend totally new items. Consider that, for instance, two different items with the same features, have also the same probability to be suggested. The other approaches, instead are strongly biased by the popularity of specific items. This holds also for ALMM. In fact, even though it exploits item features, it strongly depends on the observed frequencies of consumption of items.

5 CONCLUSION

Motivated by the task of supporting tourists in finding interesting and relevant POIs to visit we have introduced a novel recommendation approach that is based on a general user behaviour model. The model is learnt relying on the knowledge of a set of trajectories formed by sequences of POI visits in physical areas. These trajectories are clustered and taken as input for solving an Inverse Reinforcement Learning problem which determines, for each cluster, the reward function and the optimal POI selection policy. The learnt behavioural model explains the preferences of the users belonging to a cluster by identifying the relative contribution to the reward given by the various features that describe the POIs and the context. The proposed recommendation strategies adapt the next visit-action recommendations to the learned model, and maximise the reward the user gains while discovering relevant, novel and non-popular POIs. We evaluated the method with an off-line experiment using real world data about tourist visits in the historic centre of Florence (Italy).

Further research needs to examine more closely how contextual and temporal aspects, e.g., crowdedness of POIs, can be considered in the user behaviour model in order to better tailor the recommendations. Moreover, a user study aimed at comparing predicted with actual user satisfaction is in order.

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. Chou, Y. Yang, J. R. Jang, and Y. Lin. 2016. Addressing Cold Start for Next-song Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*. 115–118.
[4] D. Greene, D. O'Callaghan, and P. Cunningham. 2014. How many topics? Stability analysis for topic models. In *Machine Learning and Knowledge Discovery in Databases*, Vol. 8724 LNAI. 498–513.
[5] N. Hariri, B. Mobasher, and R. Burke. 2012. Context-aware music recommendation based on latenttopic sequential patterns. In *Proceedings of the 6th ACM conference on Recommender systems - RecSys '12*. 131.
[6] 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.
[7] X. Jia, F. Sun, H. Li, Y. Cao, and X. Zhang. 2017. Image multi-label annotation based on supervised nonnegative matrix factorization with new matching measurement. *Neurocomputing* 219 (2017), 518 – 525.
[8] D. D. Lee and H. S. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.
[9] 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.
[10] O. Moling, L. Baltrunas, and F. Ricci. 2012. Optimal radio channel recommendations with explicit and implicit feedback. In *Proceedings of the 6th ACM conference on Recommender systems - RecSys '12*. 75.
[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] G. Neu and C. Szepesvári. 2009. Training parsers by inverse reinforcement learning. *Machine Learning* 77, 2-3 (2009), 303–337.
[13] 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.
[14] E. Palumbo, G. Rizzo, and E. Baralis. 2017. 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*. 1–8.
[15] D. Ramachandran and E. Amir. 2007. Bayesian inverse reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence 2007 - IJCAI'07*. 2586–2591.
[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] G. Shani, D. Heckerman, and R. I. Brafman. 2005. An mdp-based recommender system. *Journal of Machine Learning Research* (2005), 1265–1295.
[18] R. S Sutton and A. G. Barto. 2014. *Reinforcement Learning: An Introduction (Second edition, in progress)*. The MIT Press.
[19] S. Vargas and P. Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 5th ACM conference on Recommender systems - RecSys '11*. 109.
[20] 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.