Exploiting Explainable and Generalized Human Behaviour Models for next-item Recommendations.

David Massimo

Faculty of Computer Science
Free University of Bolzano-Bozen

This dissertation is submitted for the degree of
Doctor of Philosophy

October 2019

Table of contents

Chapter 1

Introduction

In this chapter we outline the motivation and the purpose of this thesis. We present the research questions and hypotheses that steer this study. Then, we abridge the contributions and report the thesis structure.

1.1 Motivation

Recommender Systems are software tools aimed at easing users' decision making [56]. These systems help users in identifying information that most likely match their interests by means of suggestions that are generated by mining their behaviour data, e.g., ratings or clicks. Applications of Recommender Systems can be mainly found in web scenarios where there is a large amount of information that easily brings the user to a situation of information overload. For instance, a user who has to buy goods on Amazon, rather than browsing the 350 million items Amazon catalogue, can be suggested with a subset of items (from the catalogue) that she can likely appreciate.

Nowadays, in the era of Ubiquitous computing, people are constantly connected to the internet via their mobile devices and move through environments that are augmented with sensors, e.g., smart cities. Therefore, user's behaviour data are fast pace generated and give us diverse information: these data not only tell us, e.g., about a user's purchased goods on an e-commerce platform, but also describe how the user interfaces herself with the surrounding physical world. Given the different type of user's behaviour information, in this thesis we distinguish between online and offline behaviour data.

Online user's behaviour data are essentially the source of information used as input in Recommender Systems, i.e., user's actions on the web. For instance, online data may consist of the ratings given by a user to movies on streaming platforms. These ratings express the utility that the user perceives to obtain by consuming the item. Beside, online data may also be the clicks on the skip button of a media player that can be interpreted as a sign that the user didn't complete the consumption of the media content because she is not happy with it. Similarly, user's actions performed in a physical environment, e.g., staying in a place, can be exploited as proxy of the user preference.

Indeed, offline user's behaviour data consist of records of individual's physical actions like moving between two locations where user's movements data may be recorded by the sensors on the user mobile, e.g., the GPS. Moreover, offline data may consist of payment data, that can be recorded by the user's adopted payment system, e.g., credit card, or of data collected through sensor augmented spaces. These spaces can be, e.g., shops where the shelves are equipped with IoT sensors that records which customer picked a specific product.

Although a lot of remarkable progress has been made to provide high quality recommendations to users on the web, little has been done to support users in physical spaces. A possible motivation for this can be found in how Recommender Systems, and in general information systems, have been thought: a user interacts with a system from her computer. Hence, a user has been seen as being able to interact with the virtual world via her browser from a fixed position. So, any possible interaction with the real world was neglected.

In this thesis, we aim at paving the bridge between the current objective of Recommender Systems, that is the support of users in the online virtual environments, with the novel scenario, made possible by the advancement in sensing solutions, that consist of supporting users interacting in the physical space (offline environments). Additionally, we seek to ways to exploit the different kind of behavioural data (online and offline) to generate interesting recommendations for users that are acting in any of the two dimensions. For instance, a tourist, can be suggested with Points of Interest (POIs) to visit by exploiting her observed online behaviour, e.g., her bookmarks about activities in Florence in TripAdvisor, while she is visiting the city of Rome.

1.2 Research question and hypotheses

In this section we present the research questions and hypotheses that motivate the research presented in this thesis.

How to proper model user behaviour in sequential decision making? In particular, how to design and learn the behavioural model when the user is interacting with a physical environment? (RQ1)

Influencing factors of user's decision making are the context in which the user operates and the order in which items are proposed to the user. To tackle these two issues Context-

Aware Recommender Systems [2] and sequence-mining methods [48, 38, 25] have been devised by Recommender Systems researchers.

Motivated by the willingness to better understand users preferences in different contexts and the influence of the presentation or consumption order of items, we have to identify a suitable model that can learn the user preferences from her action observations.

Deriving users' preferences from their actions observations in physical spaces is not trivial because one may encounter three different issues. Indeed, observation data of user-space interactions are recorded by sensors and are therefore affected by measurement errors. For instance, if the strength of the GPS signal is weak, the uncertainty of the measured location may be in range of dozens of meters. As a consequence, we may confound the user's locations and then assume that she visited the wrong place. Moreover, users acting in an open environment, e.g, tourists, are not constrained (guided) by a fixed navigation scheme like the links on a web page. Therefore, they may stuck moving around the same places, e.g., popular attractions. This leads to observation data describing a suboptimal behaviour. Finally, the fact that users are not willing to share much about their personal (and especially location) data [51] entails a lack of available databases that contain large number of individuals' user-space interaction data. Therefore, there is the need to identify a user behaviour learning solution that can cope with scarce user's observed behavioural data. This question is addressed in Chapter 3 and 6.

How can we use the learnt user behaviour model to generate more effective recommendations (RQ2)?

Most of the current RS approaches do not distinguish between user behaviour learning and the recommendation generation process. Current RS techniques, e.g., sequence-mining [38, 25], recommend items by identifying a user's choice pattern that is directly used to identify the set of items the user is going to consume next. These are the items used for recommendation and often these recommendations are evaluated as too obvious for the target user. Therefore, we have to identify recommendation strategies that can be used in order to increase the user satisfaction rather than suggesting what the user is predicted to consume next. We focus on how to design such strategies so that they can be leveraged in combination with a learnt user behavioural model to generate recommendations that are of interest for a user. This problem is tackled in Chapter 4.

Which are the factors that make a recommendation interesting for a user? (RQ3)

Connecting with the previous research question, this one emphasizes the understanding of what makes a user choosing an item. We conjecture that there are two levels that influence the user decisions, namely a user level and an item level. In particular, at the user level we think that influencing factors are the user intent, i.e., what is the goal that drives the user during her decision making process. In addition, at the user level we argue that the user's knowledge about the domain in which she is making choices influences her interest for an item. In particular we think that users with different levels of knowledge assesses items differently: users with an higher knowledge are able to assess a broader set of items among the available options than those users with a lower knowledge.

Concerning the item level, specific item characteristics are here seen as the influencing factors steering the interest for an item. Therefore, we think that by identifying the users' intent and the set of item features in the specific domain in which the user decision making process takes place, it is crucial to gain the insight to better design Recommender Systems that are capable to generate interesting recommendations to the user. This research question is investigated in Chapter 4.

Is it possible to transfer the behaviour learnt from user-sapce interaction observations collected in an environment (source) to another one (target)? How do recommendations generated by leveraging the transferred user behaviour model compare with recommendations generated by using the true observations in the source environment? (RQ4)

This research question strictly relates with the well known problem of cold-start, i.e., few or no information about users' preferences or consumed items. In the case of user-space interactions in the physical world, e.g., a user that visits a city for the first time, the lack of users' specific information about her preferences is a common problem. Among the solutions devised to alleviate the cold-start problem, Cross-domain Recommender systems [11] have been proposed. A Cross-Domain Recommender System tries to build an encompassing user preference model by mining users' preferences about items specific to a domain, e.g., books, in order to exploit this user model in other domains, e.g., movies. We argue that by exploiting observations of user behavioural data, collected in both online and offline environments, would help to build a generalized behaviour model that can fill the gap between the two types of environments (virtual and physical) and the related user behaviour. For instance, online collected behavioural data about visitors preferred attractions in Florence can be used to learn a generalized user behaviour model to be employed for two objectives. At first, a new generalized behaviour model can be learnt for a new city (Rome). Then, POI-visit recommendations can be generated for users that are visiting Rome. Thus, we have to understand if there is a common set of features that can be leveraged to: (1) describe more than one physical space; (2) learn a generalized user behaviour model in one physical space and transfer the acquired knowledge to another (new) physical space. Another aspect that we aim at inquiring is how this transfer of knowledge affects the recommendations.

1.3 Contribution

Here we summarize the main contributions of this thesis. Each listed item is the product of the research work steered by the research questions discussed in the previous section.

User behaviour learning from user-space interaction observations. Concerning the first research question (RQ1) we have proposed a novel approach that accomplishes the following goals: (1) it exploits both online and offline user behavioural data; (2) it deals with situations where users' data is scarce and there is no additional information about users apart from their past observed actions. As application domain of our user behaviour modelling approach we selected tourism for two main reasons: firstly, it is a natural scenario in which user interacts with the surrounding environment, e.g., visiting a place or eating in a restaurant; then, users while visiting a place are in a constant sequential decision making loop, e.g., the user plans her activities and may also have to change them on-site. In particular, we operationalized the user behaviour modelling and learning approach in two scenarios: open and closed spaces. User-space interactions in the two aforementioned scenarios are acquired either from sensors, e.g., IoT augmented spaces, or from online platforms, e.g., social media. Experimental results have shown that by employing our modelling and learning appraoch the learnt user behaviour generalizes the true user preferences. The proposed solution can learn even in situation of scarce behavioural data by clustering like-behaving users and learning different generalized user behaviour model (one per cluster). This is possible due to the fact that learning is performed by generalizing over a set of features describing the context and the physical space in which the users' performed their actions.

Devising and evaluating Recommendation Strategies that harness a user behaviour model. From the knowledge acquired from the user behaviour modelling and learning we have devised recommendation strategies that leverage a generalized user behaviour model. In particular, recommendations can be generated for each user behaviour model that is learnt for a cluster of like-behaving users identified in the data. Thus, different segments of users may be supported with ad-hoc suggestions. Our main contribution is to shape a class of recommendation models that by better decoupling user behaviour learning from the generation of the recommendations, provisions to a target user recommendations that do not follow his predicted behaviour. This is an aspect that have been neglected so far: most of the current Recommender System generates recommendations reproduces the user behaviour rather than supporting her in the discovery of items that she unlikely will find without the the help of the Recommender System. This can be related to the fact that systems are evaluated and hence optimized by means of the fictional train and test split of user observed data. Offline experiments showed that the suggestions generated by our recommendation strategies provision the user with recommendations that are less accurate, in the sense that they deviate from the observed user's actions in the test data, but let the user to identify interesting (new) items that also increase the overall utility (satisfaction) of the user.

Our next contribution is to inquiry the users' perception of the recommendations generated by the recommender models we devised. Specifically we have conducted and analysed the outcomes of a user study. The user study design considered the user known items in a set of POIs and the evaluation of recommendations generated by our recommender models and baselines. The experimental results showed that our models are able to generate suggestions that are interesting POIs that are unknown to the user. This result indicates that these models may better accomplish the main task of a recommender system (in domains like tourism).

Building an application that supports users in the identification of places in the physical space. Our next contribution is related to provide an answer to the three research questions (RQ1, RQ2 and RQ3). With the lessons learned from the user behaviour modelling and learning as well as the design and evaluation of recommendation strategies that leverage the learnt (generalized) behaviour model, we have built a mobile application that supports users in identifying relevant places in the physical space. This is necessary in order to test the models integrated in a system accessed by users while they interacts with the surrounding space. The app helps tourists and locals in identifying POIs in a delimited area in North Italy app has been conceived to collect both the user's explicit (e.g., opinions) and implicit feedback (e.g., location). Thus, by using this app we can analyse the relations between the online and offline behaviour of users. A user can provide (online) her feedback on multiple levels: she can bookmark POIs that she finds relevant and want to visit during her travel; she can express an opinion about a POI; she can debrief her visits to POIs (user-space interactions). The offline behaviour of the user is collected by background processes that updates the user location by either using the mobile GPS sensor or interacting with an IoT network that augments the surrounding environment.

Transfer of users' learnt behaviour. With regard to the research question (RQ4) we have shown that by leveraging the user behaviour learnt in one city, by exploiting users' POI-visits reconstructed from an online platform, it is possible to generate recommendations in another city. Experimental results how the recommendation performance is affected when there are no observations that can be leveraged to learn a user behaviour model in an environment (target domain) and the recommendations are generated by exploiting the generalized behaviour model learnt in another environment (source domain). This situation is of pure cold-start. Moreover, we show how the learning of a generalized user behaviour model in an environemnt (target domain) can be bootstrapped by leveraging the one learnt in another domain (source domain).

1.4 Structure of the thesis

We herby outline how this thesis is structured.

In Chapter 2 we present the related research. In particular, we present a general overview about human behaviour modelling and prediction. Then, we introduce the state-of-the-art Recommender Systems focusing on sequential recommendations. Afterwards, we overview Inverse Reinforcement Learning and its application to human behaviour modelling. The chapter concludes with a survey of methods to collect user-space interaction data.

Chapter 3 explains the details of our approach for user's behaviour modelling and learning. In particular, in this chapter we showcase how to model users' behaviour in the context of tourism.

The recommendation strategies that supports users in finding interesting activities they can do next in the physical environment, are the subject of Chapter 4.

Chapter 5 covers the topic of transfer learning. The chapter showcase how we can learn user visit behaviour in the cities of Florence, Pisa and Rome by identifying, at first, a set of common features that describes the visit context and city own POIs. Then the chapter unveils how user's preferences over the identified set of features, e.g., learned from tourists movements data in Rome, can be leveraged to generate recommendations in another city, e.g., Florence.

In Chapter 6 we present the mobile app that have been developed to test our Recommender System approach with real users that are visiting a physical space.

Chapter 7 summarizes our findings and presents possible research aspects that branches from the current work.

Chapter 2

Related work

2.1 Human behaviour modelling

In this thesis, with human behaviour we refer to the course of actions an individual or a group of individuals take in order to complete a task, like deciding which book to read under the beach umbrella or which restaurant to reserve for a family lunch. Therefore, we discuss here the highly complex cognitive process, characterizing Human-decision making, that leads to the decision to perform an action, e.g., choosing which book to read among several possible books (the full set of choices).

Decision-making is a study subject of many disciplines, like economy, psychology, philosophy and computer science. A decision maker is a human or more generally an intelligent agent, that has a reasoning mechanism that enables it to make choices. In this thesis, we focus on humans who use information systems, specifically, Recommender Systems, and we simply refer to them as users. Typically, it is assumed that a user is rational, i.e., she is self-interested and acts in order to maximise the gain that it can obtain from its choices [28, 47]. Generally, in Decision-making it is assumed that a choice is tight to one (or more) outcome in relation to the world states. That said, in order to maximise its objective (the collection of gains), the user must know all the possible choices and the related gain. A way to model the user behaviour is to define a measure that quantifies the consequences (how much gain) of the future agent's outcome. That said, the utility tells us how much the user is or is foreseen to be "satisfied". In order to compute the utility of the user's choices a mathematical model is needed; it can be defined by an expert (e.g., a behavioural scientist) or inferred from the observations of an user's (or group of users) actions.

An expert that formulates the mathematical model that captures the underlying reasoning mechanism of a user, acts according to this three steps: knowledge extraction; model formulation; model evaluation. The process of extracting the domain knowledge can be achieved by interviewing decision-makers in the domain under investigation. Then, it follows the mathematical formulation of the reasoning mechanism. Finally, the designed decision model can be assessed by observing the deviation of the decisions performed by users and those that are the outputs of the model.

The second approach to decision-making modelling grounds on observational data of user choices (actions). Learning from data can exploit techniques developed by researchers and practitioners in the field of Machine Learning (ML). A ML model takes as input observational (behaviour) data and by leveraging statistics predicts an agent's actions. In this thesis we model human decision-making in a data-driven fashion.

2.1.1 Machine Learning for human behaviour prediction

In ML the need of domain knowledge is generally limited to the definition of the settings in which the decision-maker is acting. This is typically done by modelling the decision-maker as a vector, which elements represents meaningfully the characteristics of the decision scenario. For instance, if we want to build an agent that mimics the decision-making process of an individual that has to book an hotel for a work travel, we should consider as the vector components, features such as the travel dates and the budget the decision maker is ready to pay for his stay. The numerical values that allows to identify the hotel the user will probably decide for can be computed in several ways; this ways of identifying a choice (solution) are grouped in these areas of ML.

Supervised learning [57] models takes as input past user's choice observation data in order to model and predict future user's choices. Approximating the choices of a user is possible due to the presence of labels, that annotates the observed choice actions, in the training data. For instance, in the case of the previous hotel booking example, the training data can consist in the amount of money the user want to spend, check-in and check-out dates and the booked hotel.

Unsupervised learning [22] aims at discovering more about the data rather than approximating the user behaviour. Such models are employed when data cannot be labelled and there is the need to have a deeper insight about the underlying distribution or patterns in the data. For instance by applying clustering, we can identify specific groups of hotels' guests just by having their demographic information. Then, these groups can be further studied in order to gain more insight about their decision making process.

Reinforcement learning [62] is the ML technique that frames DM closer to how a behavioural scientist or economist see real decision-making. In Reinforcement learning a user (agent) is self-interested and acts with the intent to optimize an objective function. The environment in which the user acts is unknown to it. The user's choices are made in such a way that the cumulative reward she can obtain from its actions is maximised. The reward can be intended as the utility (or satisfaction) the user earns by acting in a specific way. In a similar way to expert-driven behaviour modelling the reward an agent collect by its interaction with the environment is defined by an expert. By employing an RL agent it is possible to map states, describing the environment conditions at the user decision time, to actions, the actual user's decision that leads to a next state. By computing how much the agent increases its cumulative reward by performing specific actions (decisions) we learn its policy, or even better an optimal policy. The optimal policy tell the agent how it should act in order to maximise its cumulative reward given its current state. When observations of human decisions are available, e.g., the POIs-visit itinerary of a user, then a policy (i.e., the human behaviour) can be learnt from the data by using Inverse Reinforcement Learning.

Applications of ML models in which human behaviour is the learning objective are, for instance: learning the health status of citizens from food purchases [3]; learning and predicting individual's lifetime expectation and health leveraging smokers behavioural data [14]; learning planning strategies of herders [15]; learning user preferences about items (e.g., movies) by observing online users' behaviour in order to generate recommendations about items they will like to consume in the future [56]. In this thesis we focus on modelling user behaviour for applications to Recommender Systems.

2.2 Recommender Systems

Recommender Systems (RSs) are software tools that aim at easing people decision making by delivering personalised suggestions about items to consume or actions to perform [56]. Nowadays, RSs are the drivers of the business of many companies in a variety of sectors, like tourism, e-commerce, video and music streaming. Examples of companies that successfully employed RSs are: in the tourism domain, Booking.com, Expedia and Tripadvisor; in E-commerce, Amazon and Idealo; in the video and music industry, YouTube and Spotify. The success of RSs in business applications is due to its dual function to serve both the company strategy and the end user's need as well. As reported in [56] examples of these functions are, from the company perspective: increase the quantity of items sold; increase the user satisfaction; increase the fidelity of the user. Instead, the end user is supported to: find good items; find relevant items; help other users to find good/ relevant items.

A RS in order to identify items that are suitable for a user needs to estimate the utility of the items in the system dataset. The typical assumption in RS is that the user wants to maximise the utility associated with an item. This is similar to what is presented in Section 2.1. In order to perform the computation of this utility a RS needs to process observation data about the user (past) behaviour, e.g., which are the items purchased by the user or what are the features that characterizes the purchased items. So, these observations may capture at different extent the information about the user actions (e.g., clicks or purchases) and the item information (e.g., description).

Given the user behaviour data a RS can employ different techniques to identify which items are relevant for a user.

2.2.1 Recommendation techniques

A classification of the methods that have been devised and deployed to generate recommendations for a target user are here presented.

Content-based RSs [36] suggest items of interest based on an analysis of their features. For instance, a media recommender system exploits the features of the media content (such as topic, length or media type) and suggests items that have the features contained by media that the user has consumed or liked before.

Demographic recommendations techniques [7] identify items to be suggested to a user by leveraging her demographic information. The motivation of this method is rooted in the idea that different type of user need different type of recommendations. For instance, a user of a Point of Interest RS may be suggested to visit the most popular places if she is a tourist, whereas in the case she is a local the recommendation can be to visit more niche places.

Knowledge-Based recommendations [16] are generated by relying on rules that establish how certain item features satisfy the user needs and preferences. From the human behaviour modelling perspective such systems are expert-driven. The system designer infer from the domain knowledge a mathematical function that computes the similarity of the user needs and the possible recommendations.

Collaborative Filtering [30] identifies items to recommend to the target user by exploiting only information pertaining to the set of items she evaluated (her actions) and those of similar users. Users are considered similar if they acts similarly, e.g., they rate higher a specific item and lower another item. Collaborative Filtering is based on two types of user behaviour data: explicit feedback data, like ratings and likes, or implicit feedback data, like clicks on a webpage. While explicit feedback is usually in a binary scale (i.e. like or dislike) or a likert-scale (star ratings), implicit feedback [24, 49, 19] recommendation techniques exploit observations of users' actions performed on the items. For instance, a RS, in order to suggest an internet video, could exploit data about browsing behaviours while watching videos, e.g., the fact that certain videos are skipped or partially seen.

Hybrid RSs [10] combine two or more of the presented recommendation strategies in order to improve the quality of the suggested items avoiding the pitfalls of a specific strategy by leveraging the benefits of another RS technique. For instance, the inability of Collaborative Filtering methods to compute the utility of an item if user-item interactions are absent in the observation data, can be overcome by exploiting the capability of Content-based methods to harness the descriptive features of the item to identify items with the same (or a subset) of features that the user (or users) consumed before.

Since the motivations behind the choices of a user may depend on external factors, that defines the decision context of the user, e.g., the time or the location dimensions in which the user is placed, Context-aware RS (CARS) have been proposed. CARSs tries to improve recommendations by considering the user preferences about items and by modelling the context in which an item has been consumed in order to predict which item will be more likely consumed by users that are in a given context [2]. CARSs have been used in various application domains like news, movies and tourism [35, 1, 9] and several context modelling techniques have been proposed [6, 8].

In this thesis in order to learn better users' action selection policies and preferences, we aim at finding a context-model that can be leveraged by recommendations techniques that generate effective sequences of recommendations to a user.

2.2.2 Sequential recommendations

Most of the approaches exploited in RSs implicitly assume that the consumption of goods and their evaluations are independent from the order in which the items have been consumed. For instance, content-based and collaborative based techniques do not exploit the user's item evaluation order to build their predictive models. In some cases, rather than predicting a set of items that could be relevant for a user, the goal is to identify items that could be consumed immediately after a specific item. For instance, during a visit in the city of Bolzano (Italy) what can be recommended to a tourist that visited, at first, Walther square and, then, an exhibition in Museion, that are two very close but different venues? In order to address this type of questions, the sequential aspect of item consumption has been studied in the context of personalisation by proposing pattern-discovery approaches in different domains: web [38], music [20, 25], travel [67, 40, 48] and cultural heritage [21].

Alternatively, Reinforcement Learning techniques have been employed in order to design a RS that considers the items' consumption order. In [59, 39] were computed optimal decision making policies to offer adapted content as a sequence of items to users, whereas in [37] Reinforcement Learning was used to compute the optimal interaction policy of a conversational recommender whose suggestions were generated with case-based reasoning.

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) that the system is supposed to obtain by showing certain information to the user in response to the user's requests. 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 [65].

In this thesis, in order to not suggest uninteresting items to users and to deal with situations in which users' explicit feedback is scarce, we propose a recommendation technique that harnesses a generalised user behavioural model that is learnt via Inverse Reinforcement Learning (IRL) by observing a group of users (cluster) "similar" to the target one.

2.3 Inverse Reinforcement Learning

To fulfil the need of learning an explainable user behavioural model from user behaviour data, imitation learning is a viable solution. It is typically addressed by solving Markov Decision Problems (MDP) via Inverse Reinforcement Learning (IRL)[44].

Environment and Decision Maker representation

A MDP is defined by a tuple (S, A, T, r, γ). S is a finite set of states (e.g., visit to a location). A is a finite set of actions (e.g., moving to a location). 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. The function $r : S \rightarrow \mathbb{R}$

models the reward a user obtains from acting in a certain way (being in a state). This function is supposed to be *unknown* and must be learnt. Finally, $\gamma \in [0,1]$ is used to discount future rewards with respect to immediate ones.

Given a MDP, our goal is to find a policy $\pi^* : S \to A$ 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_{π^*}. So, in order to compute Q_{π^*} we rewrite the previous formula as:

$$Q_{\pi^*}(s,a) = \sum_{s'} T(s'|s,a) \left[r(s) + \gamma \max_{a'} Q_{\pi^*}(s',a') \right]$$

Given a demonstrated behaviour (e.g., user actions sequences), IRL models solve the target MDP by computing the reward function that makes the behaviour induced by the optimal policy for that reward (the learning objective) close to the demonstrated behaviour.

Despite the fact that the reward is learnt from behaviour observation, the induced policy may not uniquely identified. It means that an IRL problem has many solutions. In order to overcome this problem several algorithmic strategies has been proven to identify a solution that justifies the observed behaviour. In [44] the authors suggest to introduce a margin to a linear programming formulation of the IRL problem, such that the difference between the (learnt) reward obtained from a policy and the reward of other policies is maximised. In [70] the reward function is sought by matching state features in the observation data. Maximum-entropy is used to identify actions probabilities that lead to a reward that supports the observed actions. In [5] the identification of a solution that supports the observations is identified by combining maximum likelihood estimation with a gradient method.

Learning

IRL algorithms take as input a model of the environment, the MDP, and the observed behaviour of a user (or any autonomous agent) in the form of demonstrations, in our case the state-action trajectories ζ. For instance, the trajectory ζ_u user u which is a temporally ordered list of state-action pairs (user-space interactions). For instance, $\zeta_{u_1} = ((s_{10},a_3),(s_5,a_8),(s_{15},a_e))$ represents a user u_1 trajectory starting from state s_{10}, moving to

s_5 by performing action a_3 and ending to s_{15} by acting according to a_8. The last action a_e is a dummy action that indicates the end of the trajectory. With Z we represent the set of all the observed users' trajectories. trajectories in Z, and return the inferred user's reward function. In IRL the underlying assumption is that a user is a rational decision maker who seeks to optimize the reward associated to her actions. Due to this, the agent is typically referred as "expert".

Generally the state space S is represented by a state feature function $\Phi : S \rightarrow \mathbb{R}$ that assigns to each feature a real value.

In order to infer the user's reward from her observations there is the need to identify a solution, i.e., a reward function, that makes the observed behaviour optimal. 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 [44]) that r is a linear function, $r(s) = \theta^T \phi(s)$, 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.

Researchers in the field of IRL showed that the reward estimation problem is ill-posed because there is an infinite number of solutions, e.g., a reward function $r = c$, where c is a constant, is an example of such problem. Therefore, the challenge in IRL is to seek for a solution that is optimal, the best among the set of all the solutions. The main difference among the IRL algorithms proposed in the literature is in how the solution is computed, i.e., they differ in the optimality criterion.

To resolve the issue of identifying an optimal solution in [44, 54] has been proposed to add a margin in order to maximize the difference between the reward derived from the optimal policies an the reward that is derived from the alternative policies. In [53] the authors tackle the problem of computing the reward from a Bayesian perspective. The proposed model, called Bayesian IRL, leverages the users' observations in order to infer the optimal reward. At first, it uses the observations as evidence to update the prior knowledge on the set of possible reward functions (solutions), which are assumed to be independently identically distributed. Then, Bayesian IRL estimates the reward using the posterior knowledge. The authors of Maximum-Entropy IRL [70] propose to seek for a reward function by matching state features in the observation data. Maximum-entropy is used to identify action (i.e., user-space interactions in our case) probabilities that lead to a reward that supports the observed data.

2.3.1 Maximum Log-Likelihood IRL

In this thesis, in order to learn both the user's reward and her action-selection policy, we use a specific IRL algorithm called Maximum Log-likelihood (MLIRL) [5]. MLIRL combines many positive features of other IRL models [53, 70, 42]: 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, i.e., by sampling choices (actions) from a Boltzmann distribution.

The algorithm exploits the fact that a guessed θ induces a probability distribution over action choices and hence determine a likelihood for the observations in Z. Expected values (discounted) are computed via the following formula:

$$Q_\theta(s,a) = \theta^T \phi(s) + \gamma \sum_{s'} T(s,a,s') \frac{\sum_a Q_\pi(s,a) e^{\beta Q_\pi(s,a)}}{\sum_{a'} e^{\beta Q_\pi(s,a')}}$$

MLIRL looks for $\theta = \arg\max_\theta L(Z|\theta)$ which is the maximum likelihood solution that is found via gradient ascent optimisation. The log likelihood of the observed trajectories Z is defined as:

$$L(Z|\theta) = \prod_{i=0}^{|Z|} \prod_{s,a \in \zeta_i} \pi_\theta(s,a)$$

The term $\pi_\theta(s,a)$ in the previous equation represents the Boltzmann action-selection policy, which is defined as:

$$\pi_\theta(s,a) = \frac{\sum_a Q_\pi(s,a) e^{\beta Q_\pi(s,a)}}{\sum_{a'} e^{\beta Q_\pi(s,a')}}$$

The computation of θ via gradient ascent is performed for a fixed number of steps M. At each step the $\pi_\theta(s,a)$ is computed by solving via value iteration the MDP, using the estimated reward $r(s) = \theta^T \phi(s)$.

MLIRL is known to converge to a solution in finite-horizon settings and is also known to produce a well-defined answer. The problem of the existence of multiple reward functions for which an observed trajectory is optimal in a given MDP, is solved by assigning high probabilities to observed behaviour and low probability to the unobserved. The general steps of the code are listed in algorithm 2.

Algorithm 1 Maximum Likelihood Inverse Reinforcement Learning

Input: S, A, T, γ, ϕ, $Z = \{\zeta_1, \dots, \zeta_N\}$, M, λ_t step size.
$\theta \leftarrow$ Initialize with random values;
for t=1 to M **do**
  Compute $Q_{\theta_t}, \pi_{\theta_t}$
  $L = \sum_i \Pr(\zeta_i) \sum_{(s,a) \in \zeta_i} \log \pi_{\theta_t}(s,a)$
  $\theta \leftarrow \theta + \lambda_t \nabla L$
end for
Output: θ

Learning from scarce individual's behavioural data

As we have shown in the previous section by harnessing behavioural data of an individual, i.e., user-space interaction trajectories, we can learn with IRL the user behavioural model in terms of the user's preferences θ, her reward r and the associated action selection policy π^*.

Generally, in information systems the amount of individual user behavioural data is not large for the majority of the system users. The lack of user's data becomes more evident when it comes to user-space interaction data, for which the available public datasets are not many (and sparse as well) and the only rich datasets are those owned by service providers like Google, Foursquare, and Uber. We think that from a RSs perspective, which is the focus of this thesis, using only target user (specific) behavioural data for recommendations generation for him is of scarce utility for the user: the suggested items (e.g., POI-visits) will (probably) be those that the user would choose without the help of the RS. Moreover, individual behavioural data may present a sub-optimal behaviour. e.g., a user that visits for the first time a city may simply visit the few, more accessible, places that are closer to the city main attractions. Learning a behavioural model from such observations would lead to a biased model. We think that by learning, instead, a behavioural model from observations of more visitors in the city, the resulting learnt behaviour will minimize the impact of sub-optimal behaviours that could influence some of the observed trajectories.

In order to alleviate the problems of learning from scarce user's data and minimizing the impact of suboptimal behaviours present in the data, we propose to group the user-space interaction trajectories in clusters and then to learn a "general" user behavioural model common for all the users/trajectories in a cluster.

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, e.g., the next POI visit, the users in a cluster should take in order to maximise their reward.

2.3.2 Applications of IRL methods

. In [70] 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 [63] the author applies maximum entropy IRL in order to learn pedestrian behaviour from observed traces. The learnt behavioural model is used to generate synthetic trajectories at the city level in order to conduct simulations in planning tools. The IRL-based solution outperforms a popular baseline used in the sector of mobility and transportation to generate users' movements. In [15] 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 [29] it is proposed a method for computing the reward humans get from 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.

Leveraging IRL for behaviour learning allows not only to learn users' behaviour from implicit or explicit feedback data, taking into consideration the sequential nature of the item consumption, but also enables the to predict users' preferences about items that are unknown for them. This is possible thanks to the ability of IRL models to generalize over a set of features describing the items.

2.4 Collecting human behaviour data

In the era of Ubiquitous computing, where people are constantly connected to the internet through their mobile devices, user behaviour data are generated incredibly fast. Generated data can capture the user's interactions in the virtual world, e.g., clicks on web pages, as well as the user interactions in the physical world, e.g., the user user location. In this thesis we use the term "online behaviour" to refer to the user interactions in the virtual sphere, whereas "offline behaviour" refers to the user actions in the physical world.

Online user's behaviour data are essentially the source of information used as input in RSs (Section 2.2). Summarizing, the records of a user actions on the web are distinguished in: explicit actions (feedback), like ratings given by the user to movies on a streaming platforms that are expressing the utility that the user perceive to get from the item, and implicit feedback, like the click on the skip button of a media player that can be interpreted as a sign that the user didn't complete the consumption of the media content because she is not happy with it. Similarly, users' explicit and implicit actions/feedback can be identified in the user's offline behaviour.

Offline user's behaviour consist of records of individual's physical actions like moving from a location to another location or picking an item from the shelf, as well as paying with a credit card in a shop. In general, we can say that offline behaviour is a locational data source. Here we list the most used types of offline behaviour data.

- **Call Details Records** (CDRs) are data collected by cellular network operators that keep records of mobile communications. A mobile cell, covers a portion of a geographical area with a radio-frequency, with a radius that goes from 1 to 30 kilometers, and allows to transmit mobile signals, e.g., phone calls. Cells are positioned close to each other in order to cover large areas and allow communication while the user of the network, e.g., the phone caller, is moving. CDRs data offer a compromise between the space (location) and time dimension of a user. Learning human behaviour from CDRs data has been successfully done in the context of human activity recognition for the design of activity-based travel demand models [66]. The authors in [41] exploits CDRs data in combination with external information sources, like, census, land-use and social-network data in order to enable urban planners and sociologist to measure with an automated tool the "vitality" metrics of a city.

- **GPS** data provides fine-grained location data of moving objects, e.g., people, cars and cattle, in both the space and time dimension. The typical format consist of a pair of latitude and longitude coordinates annotated with a timestamp and the accuracy of the measure. Errors on the recorded locations are in the range of few meters and each location update can be recorded even at the scale of seconds. Processing GPS datasets in order to identify precious insight about the behaviour of moving objects has been extensively studied in trajectory data mining. In [69] the authors proposes techniques for travel recommendations base on stay point detection of people's raw GPS data. GPS data are processed to identify clusters of records that, thanks to a timestamp annotation, allow to identify candidates locations where the user performed some activity. In transportation applications, rather than having the information about stay points of individuals, is important to understand which are the road segments traversed by the user and her direction. In [43] is proposed a map-matching model that can provide an high accuracy traversing behaviour of a moving object. By starting from raw GPS traces the model considers transition probabilities and the topology of the road network and identifies the actual segments traversed by the user.

- **Location Based Social Network** (LBSN) data is a rich resource that combines exact location data annotated with user feedback, like, ratings, reviews or opinions about the location. Nowadays, many deployed online systems are LBSN, these are, for instance, Foursquare (location and opinion sharing) and Waze (navigation). Given the richness of their records, LBSNs have been extensively used to model and predict human behaviour. In [27, 23] LBSN data are extensively used to analyse and explain social aspects at the urban level, like, understanding how diverse groups of citizens interacts with the surrounding space. The deep level of understanding about the places where a user spent time and her opinion of them comes at the cost of having very sparse data. LBSNs users have to explicitly insert data records in the system and therefore, as it usually happens in information systems, the user feedback is rather scarce. This makes the usage of check-in data for sequential decision-making inference, like deciding which route or the next location to visit, more difficult than GPS data.

- **Internet of Things** (IoT) is a dynamic and global adaptive network assisted by an intelligence that coordinates the communication between the connected devices, the so called *things*, in order to achieve a goal [34, 33, 4]. IoT relies on the following technologies: radio-frequency identification (RFID), Near Field Communication (NFC), low energy wireless communication (i.e. beacon) and Wireless Sensor Networks (WSN) which are networks that connects these sensors via wireless communication. IoT enables to collect diverse type of feedback from humans. For instance, by placing IoT devices in an environment it is possible to respond to users' actions in real time as well as collecting data about their behaviour [31, 46, 50]. Human-IoT interactions can be enriched with location data, e.g., using the network or GPS on the user's mobile or by using the IoT device location. Hence, this type of data can provide both implicit and explicit information about the user activities that can be leveraged to analyse and learn human behaviour.

In order to collect users' actions data and contextual information to learn users' preferences and behaviour, in this thesis we discuss a learning solution that exploits and combines as input sources: users' GPS traces; users-IoT interactions in IoT augmented spaces; external web repositories like LBSNs. Due to the nature of the data, observations of actions for which the user's reward is unknown, we base the learning solution on IRL. The objective of designing RS technologies that harness the users' learnt behaviour is achieved by: setting up a proper context-model that can capture the decision situation of a user; identifying behaviour and preference patterns of the users by means of unsupervised learning, i.e., clustering; exploiting, as in Content-based RSs, the information about the user's consumed items, e.g., visited places.

Chapter 3

User preference and behaviour learning in physical spaces

3.1 User-space interaction

The underlying idea that motivates the research presented in this thesis is that RSs technologies can be employed not only to support people when they interact in the virtual world, i.e., the web, but also when they act in physical environments, i.e., a city. This is possible due to technological advancement in the field of sensing solutions, that brought novel possibility to capture human behavioural data in real environments, i.e., recording the offline user's behaviour. Sensed user behavioural data can then be leveraged to learn user's preferences. In this thesis we mainly focus on behavioural data acquired from sensors like GPS and IoT devices, such as, beacons.

3.1.1 GPS data

The GPS sensor on the mobile device of a user provides fine-grained location data that describes the mobility behaviour of the user. This information is typically formatted as a tuple $g = (lat, lon, t, \mu)$, where lat and lon are the latitude and longitude of the sensed location, t is a timestamp and μ is the accuracy of the measure. A GPS device can update the location at the scale of seconds and can have an error in the range of few meters. GPS data with different configurations of location updates and accuracy are used to support users in different scenarios: short interval location updates and high accuracy measures are typically used in navigation applications where the goal is to drive a user from location A to B in real time; non-frequent and less accurate locations updates are used in LBSNs applications in order to identify locations of interest for a user.

GPS data are by nature noisy and therefore need to be processed in order to be used in an application. The field of study that aims at extracting insights from GPS traces is trajectory data mining. The term “trajectory” indicates the fact that a GPS device generates a trajectory $\zeta_{GPS} = (g_j : j \in \{0,\ldots,n\})$ composed of n location updates. In this thesis we adopt the following trajectory data mining techniques: stay point detection; trajectory segmentation; trajectory clustering; map matching. Stay point detection techniques are employed to identify the location, within a certain radius, where a user, or any moving objects, stayed for a given time-interval. A stay point can be, e.g., a restaurant or a museum that a user has been to, and, in addition to the GPS locations in a trajectory, it carries a deeper (semantic) meaning (i.e., it describes the user’s action). Trajectory segmentation methods deconstruct a trajectory into sub-trajectories by time interval, spatial shape, or semantic meanings. This representation is generated before performing clustering or classification. Map matching techniques aim at projecting trajectory GPS location onto the corresponding road segment where the point was generated.

The details of the trajectory data mining techniques that we employ/designed in order to process GPS data are detailed in Chapter 6.

3.1.2 IoT data

IoT technologies make possible the exploitation of sensors networks to enable new ubiquitous information services [34, 33]. In fact, by distributing sensors in an environment or even by integrating them into objects it is possible to respond to user actions in real time as well as collecting data about the user behaviour [31, 46, 50]. The IoT sensors that we consider for collecting human behavioural data are those that exploit RFID, NFC and BLE short-range wireless technology. In contrast to GPS sensors, BLE allows to capture the actions a user performs as well as their semantic.

Peculiar to IoT augmented scenarios is the possibility to collect user physical actions data not only outdoor, but also indoor. For instance, by augmenting a physical space like the exhibition room of a museum (indoor) or a square in the old town of a city (outdoor) with a beacon device, i.e., small BLE devices broadcasting low-energy Bluetooth messages encoded with standard transmission protocols (e.g. Eddystone or iBeacon), is possible to collect user’s behavioural data. The broadcasted messages can be sensed by the Bluetooth receiver of the user mobile (smartphone) and, with the aid of background processes running on the devices, can fire the generation of location-based notifications or feed information to a user model in order to support further personalization of the system generated information [45]. In addition, IoT augmented objects enable new possibilities to collect behavioural data about the user-space interactions. Sensors enabled objects allow to detect when they are moved and manipulated. This enable the possibility to design interactive scenarios where descriptive information about objects is presented to users at the very exact time when they are inspecting them, hence, stimulating enjoyment and sharing [58].

We represent an interaction of a user with an IoT augmented place or object as a tuple $i = (id, a, t)$. With *id* we denote the identifier of the IoT device with which the user interacted. The action *a* performed by the user represent the semantic of the physical action, e.g., with "visit" we represent the visit to a POI or with "play" we mean the fact that a user started a media content. With *t* we model the time (timestamp) at which the action *a* is performed. Specific of user-IoT interactions is how the (geo) location information is handled: the *id* of the IoT device can be used to enrich the record *i* with information about the user location by using application domain knowledge, i.e., IoT devices are deployed in fixed positions of specific areas. Alternatively, the GPS of the user mobile device can be leveraged to annotate with the location coordinates the sensed interaction. The IoT traces of a user that who interacted *n* times with the physical environment form a list $\zeta_{IoT} = (i_j : j \in \{0, \ldots, n\})$ composed of actions updates (and locations) *i*. In order to get more information about technical aspects about the IoT infrastructure that we have designed, in order to trace and respond to user's actions in sensor enabled spaces, we refer to our study "Tangible Tourism with the Internet of Things" [12].

3.1.3 Social Network data

Scientists in the fields of urban computing and computational social science have investigated how user behavioural data can be derived from social networks in order to investigate mobility and socio-economic aspects in specific geographical areas [68, 61, 60].

LBSNs offer rich information about users' interactions in the physical space. For instance, in photo sharing platforms like Instagram[1] each photo provides additional insight (e.g., descriptive tags, likes) about a location (geo coordinates) at a given time, whereas in check-in platforms like Foursquare City Guide[2] a location (e.g., a POI) is enriched with metadata like the POI category (bar or shop) and opinions of the users (ratings or reviews). Even though LBSNs offer such level of information about specific places in the physical environment, user's data are generally sparse if compared to the amount of data a GPS sensor can collect.

Besides LBSNs, more traditional social networks, such as the photo sharing platform Flickr, offer the possibility to collect user behavioural data. For instance, in [52] Flickr[3] photos, their related geo data and tags have been used to identify indoor activities in the cities

[1]https://www.instagram.com/
[2]https://foursquare.com/city-guide
[3]https://flickr.com

of New York and London. In other works [17] Flickr data has been used but never considering individual photos. In [13] Flickr data have been leveraged in order to automatically generate visit itineraries.

In this thesis we leverage Flickr data because its geo-localized pictures and their metadata are more likely to be related to the place where they have been taken. Moreover, since we are interested in learning users' preferences as well as their sequential decision making to generate next-item recommendations, we leverage Flickr data to retrieve individual sequences of observations.

For a specific user of a social network or LBSN a record can be represented as a tuple $l = (lat, lon, F, t)$, where lat and lon are the latitude and longitude of a location, F is the set of features characterizing the location, e.g., the category of a POI or aggregate feedback expressed by the community on the POI, t is the time at which the user added content to the LBSN platform. For a LBSN user is possible to build a trajectory of the n locations he was physically present $\zeta_{LBSN} = (l_j : j \in \{0, \ldots, n\})$.

In this thesis we collect user behavioural data from the Flickr platform. In particular, from photo albums uploaded by users on Flickr, where each photo is geo-tagged, we reconstruct the itinerary a user followed.

3.2 Making sense of user-space interaction data

Either we have user-space interactions (trajectories) that have been collected by means of GPS sensor on the user mobile; sensed by the users' mobile Bluetooth receiver (interaction with a Beacon); or reconstructed from the user's profile on a social network, there is the need to build a representation of each interaction with the environment that allows to infer the underlying factors (preferences) that motivates the user's (offline) behaviour.

To attain this goal we need to employ a feature representation that a ML model can exploit to learn and generalize from the data. We think that for any scenario in which a user performs decision making there are two main types of information that need to be considered: context information, describing what are the conditions in which the user operated; content information describing the items subject to the user's choices.

Let assume that any data trajectory ζ_{GPS}, ζ_{IoT} and ζ_{LBSN} of length n can be represented by a more general trajectory $\zeta = (o_j : j \in \{0, \ldots, n\})$ where the user-space interaction observation $o = (lat, lon, t)$ models the fact that an interaction happened at the location defined by the latitude and longitude pair (lan, lon) at time t. With O we denote the set of all the user-space interactions o. Let E_{ctx} be the set of contextual informations in an external resource, e.g., the content of a weather API, and let E_{cnt} be the set of content information that can be obtained from an external resource, e.g., Wikipedia. That said, in order to enrich with context and content data the observations in O, we have to identify two mappings: $\psi_{ctx} : O \rightarrow E_{ctx}$ that maps a user-space interaction o to a specific context, e.g., a POI-visit is mapped with its weather conditions; the mapping $\psi_{cnt} : O \rightarrow E_{cnt}$ that maps the same user-space interaction o with content information, e.g., a POI-visit is mapped to its category.

The external information resources to be employed in order to enrich the user-space interactions observations can be: (1) generated (or defined) by domain experts; (2) identified among available online resources, e.g., Wikipedia. For instance, in order to obtain content information in the tourist domain, with the objective of learning tourists' preferences, Wikipedia and Tripadvisor can be used as external resources. With regard to context information, it is possible to derive relevant features directly from the user-space interaction data. For instance, the crowdedness of a place can be inferred from the geo-coordinates and the time recorded in the data: by defining a bounding geographic area, all the users that interacted at a specific time in that area provides the information about the size of the crowd. For other type of context data, like the weather, online resources can be used.

Here we describe an example showing how we add, by using Wikipedia data, content and context information to trajectories of visited locations in a city. Content data falling within a geographic (bounding-box) area, defined by the minimum and maximum values of the (lat, lon) pairs in the data, is retrieved from the external information source. Then, the retrieved content is processed to identify a set of features, e.g., the place name and its type (bar or shop), to represent the location. Afterwards, each location in a trajectory can be enriched with the identified content. In this way a pair of geographical coordinates becomes a recognizable POI and the trajectory becomes the itinerary of POI-interactions the user made in the physical space. From such, richness of information in the data user behaviour information can be learnt.

3.3 Learning a user preference model

In this section we detail how we learn user's preferences and behaviour from observed user-space interaction trajectories. In particular, we present how we model the problem of the trajectory generation task, which is closely tight to the problem of sequential decision making. Afterwards, we detail how we learn user' preferences as well as her action-selection policy.

3.3.1 Problem modelling

We model the user-space interaction trajectory generation task as a finite Markov Decision Process (MDP). A MDP is defined by a tuple (S, A, T, r, γ). With S we denote a finite set of states and in our scenario a state represents the interaction of a user with the physical space (e.g., visiting a POI) in a specific context (e.g., weather, temperature and day). For instance, a tourist that visits the old town of Florence can be at the Battistero (POI) in a cloudy, cold morning (context). A is a finite set of actions, which in a tourism scenario can represent the decision to move to a POI. With T we indicate 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 Florence during 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 $r : S \to \mathbb{R}$ 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 her interaction with the environment (the user is not supposed to reveal it). But, we assume that if the user performed an action and not another one, then she believes that the first action 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 state-action pairs (user-space interactions). For instance, $\zeta_{u_1} = ((s_{10}, a_3), (s_5, a_8), (s_{15}, a_e))$ represents a user u_1 trajectory starting from state s_{10}, moving to s_5 by performing action a_3 and ending to s_{15} by acting according to a_8. The last action a_e is a dummy action that indicates the end of the trajectory. With Z we represent the set of all the observed users' trajectories. Given a MDP, our goal is to find a policy $\pi^* : S \to A$ 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_{π^*}. So, in order to compute Q_{π^*} we rewrite the previous formula as:

$$Q_{\pi^*}(s,a) = \sum_{s'} T(s'|s,a) \left[r(s) + \gamma \max_{a'} Q_\pi(s', a') \right]$$

The problem of computing the optimal policy for a MDP is solved by Reinforcement Learning algorithms [62].

As we mentioned earlier, in a information systems and specifically in RSs applications the reward obtained by a user when she is in a specific state (i.e., the r function) is usually unknown because users scarcely provide feedback (e.g., ratings or reviews about the consumed items). 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).

3.3.2 IRL

IRL algorithms take as input a model of the environment, the MDP, and the observed behaviour of a user (or any autonomous agent) in the form of demonstrations, in our case the trajectories in Z, and return the inferred user's reward function. In IRL the underlying assumption is that a user is a rational decision maker who seeks to optimize the reward associated to her actions. Due to this, the agent is typically referred as "expert".

Generally the state space S is represented by a state feature function $\Phi : S \to \mathbb{R}$ that assigns to each feature a real value. We model each state by using the features identified with the mappings Ψ_{cnt} and Ψ_{ctx} (Section 3.2).

In order to infer the user's reward from her observations there is the need to identify a solution, i.e., a reward function, that makes the observed behaviour optimal. 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 [44]) that r is a linear function, $r(s) = \theta^T \phi(s)$, 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.

Researchers in the field of IRL showed that the reward estimation problem is ill-posed because there is an infinite number of solutions, e.g., a reward function $r = c$, where c is a constant, is an example of such problem. Therefore, the challenge in IRL is to seek for a solution that is optimal, the best among the set of all the solutions. The main difference among the IRL algorithms proposed in the literature is in how the solution is computed, i.e., they differ in the optimality criterion.

To resolve the issue of identifying an optimal solution in [44, 54] has been proposed to add a margin in order to maximize the difference between the reward derived from the optimal policies an the reward that is derived from the alternative policies. In [53] the authors tackle the problem of computing the reward from a Bayesian perspective. The proposed model, called Bayesian IRL, leverages the users' observations in order to infer the optimal reward. At first, it uses the observations as evidence to update the prior knowledge on the set of possible reward functions (solutions), which are assumed to be independently identically distributed. Then, Bayesian IRL estimates the reward using the posterior knowledge. The authors of Maximum-Entropy IRL [70] propose to seek for a reward function by matching state features in the observation data. Maximum-entropy is used to identify action (i.e., user-space interactions in our case) probabilities that lead to a reward that supports the observed data.

3.3.3 Maximum Log-Likelihood IRL

In this thesis, in order to learn both the user's reward and her action-selection policy, we use a specific IRL algorithm called Maximum Log-likelihood (MLIRL) [5]. MLIRL combines many positive features of other IRL models [53, 70, 42]: 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, i.e., by sampling choices (actions) from a Boltzmann distribution.

The algorithm exploits the fact that a guessed θ induces a probability distribution over action choices and hence determine a likelihood for the observations in Z. Expected values (discounted) are computed via the following formula:

$$Q_\theta(s,a) = \theta^T \phi(s) + \gamma \sum_{s'} T(s,a,s') \frac{\sum_a Q_\pi(s,a) e^{\beta Q_\pi(s,a)}}{\sum_{a'} e^{\beta Q_\pi(s,a')}}$$

MLIRL looks for $\theta = \arg\max_\theta L(Z|\theta)$ which is the maximum likelihood solution that is found via gradient ascent optimisation. The log likelihood of the observed trajectories Z is defined as:

$$L(Z|\theta) = \prod_{i=0}^{|Z|} \prod_{s,a \in \zeta_i} \pi_\theta(s,a)$$

The term $\pi_\theta(s,a)$ in the previous equation represents the Boltzmann action-selection policy, which is defined as:

$$\pi_\theta(s,a) = \frac{\sum_a Q_\pi(s,a) e^{\beta Q_\pi(s,a)}}{\sum_{a'} e^{\beta Q_\pi(s,a')}}$$

The computation of θ via gradient ascent is performed for a fixed number of steps M. At each step the $\pi_\theta(s,a)$ is computed by solving via value iteration the MDP, using the estimated reward $r(s) = \theta^T \phi(s)$.

MLIRL is known to converge to a solution in finite-horizon settings and is also known to produce a well-defined answer. The problem of the existence of multiple reward functions for which an observed trajectory is optimal in a given MDP, is solved by assigning high probabilities to observed behaviour and low probability to the unobserved. The general steps of the code are listed in algorithm 2.

Algorithm 2 Maximum Likelihood Inverse Reinforcement Learning

Input: S, A, T, γ, ϕ, $Z = \{\zeta_1, \dots, \zeta_N\}$, M, λ_t step size.
$\theta \leftarrow$ Initialize with random values;
for t=1 to M **do**
  Compute Q_{θ_t},π_{θ_t}
  $L = \sum_i \Pr(\zeta_i) \sum_{(s,a)\in\zeta_i} \log \pi_{\theta_t}(s,a)$
  $\theta \leftarrow \theta + \lambda_t \nabla L$
end for
Output: θ

3.4 Learning from scarce individual's behavioural data

As we have shown in the previous section by harnessing behavioural data of an individual, i.e., user-space interaction trajectories, we can learn with IRL the user behavioural model in terms of the user's preferences θ, her reward r and the associated action selection policy π^*.

Generally, in information systems the amount of individual user behavioural data is not large for the majority of the system users. The lack of user's data becomes more evident when it comes to user-space interaction data, for which the available public datasets are not many (and sparse as well) and the only rich datasets are those owned by service providers like Google, Foursquare, and Uber. We think that from a RSs perspective, which is the focus of this thesis, using only target user (specific) behavioural data for recommendations generation for him is of scarce utility for the user: the suggested items (e.g., POI-visits) will (probably) be those that the user would choose without the help of the RS. Moreover, individual behavioural data may present a sub-optimal behaviour. e.g., a user that visits for the first time a city may simply visit the few, more accessible, places that are closer to the city main attractions. Learning a behavioural model from such observations would lead to a biased model. We think that by learning, instead, a behavioural model from observations of more visitors in the city, the resulting learnt behaviour will minimize the impact of sub-optimal behaviours that could influence some of the observed trajectories.

In order to alleviate the problems of learning from scarce user's data and minimizing the impact of suboptimal behaviours present in the data, we propose to group the user-space interaction trajectories in clusters and then to learn a "general" user behavioural model common for all the users/trajectories in a cluster.

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, e.g., the next POI visit, the users in a cluster should take in order to maximise their reward.

3.4.1 Clustering like-behaving users

Clustering the trajectories is implemented with Non Negative Matrix Factorization (NMF) [32], which is a specific class of Matrix Factorization models. Matrix Factorization has the objective of reducing an input matrix into its constituent parts in order to ease the computation of more complex matrix operations or inspect the input data.

Applications of NMF can be found in many field of science, e.g., in astronomy, NMF is used to process space observation data in order to identify planets that cannot be directly observed due to the high amount of light that stars close to the planet emits [55]; in biology, NMF has been used to cluster gene expression [64]. Of our interest is the application of NMF in text mining where NMF allows to group documents to a common semantic structure that can explain the resulting clusters.

NMF requires as input a positive real valued matrix, therefore documents needs to be represented by using an appropriate statistic, i.e., term frequency–inverse document frequency (tf-id). Given a set of documents, i.e., a text corpora, the tf-idf statistic represent numerically how important is a term in a document. Let $d \in C$ be a document belonging to the corpus C and let be $t \in d$ a term of the document. The tf-idf is computed by means of the following formula:

$$tfidf_{d,C}(t) = tf_d(t) \cdot idf_C(t)$$

The term $tf_d(t)$ is the term frequency of the term t for the document d; we compute it as $tf_d(t) = \frac{count_d(t)}{|d|}$. The numerator $count_d(t)$ is the number of terms in d that are equal to t.

The second term in the formula is the inverse document frequency $idf_C(t)$ and express how much a word is important, i.e., a word is rare or common in the corpora C. We compute it as:

$$idf_C(t) = log\frac{|C|}{|d \in C : t \in d|}$$

In order to use NMF to identify like-behaving/minded users from their user-space interactions trajectories we need to build a document-like representation of the trajectories. To generate such representation we harness (for each trajectory ζ) the mappings Ψ_{cnt} and Ψ_{ctx} that we defined in Section 3.2. We recall that these mappings identify descriptive features for each user-space interaction in the data. For instance, if the user visited a museum the descriptive features associated to the interactions can be: content information describing the visited place, e.g., the type of museum (science), the exhibition style (interactive); context information, e.g., the part of the day (afternoon), the weather (rainy) and the crowdedness of the place. By representing each observed user-space interaction with its associated features (terms), we generate a document that describes the observed interaction of the user with the environment. When this operation is performed by using all the trajectories in the database we obtain the corpora that describes the interactions of all the users.

With D we denote the tf-idf matrix representation of the obtained corpora. Columns in D represents specific terms and rows corresponds to trajectories. The matrix D has size $|Z| \times F$, where F is the number of unique terms in the corpora. 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) 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 a threshold τ as similarly done in [26]. 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 order to identify the correct number of topics (clusters) we conduct a stability analysis, as suggested in [18]. The procedure seeks for the best number of topics k from a pre-defined space. At first, a reference k-topic model M^{ref} is generated by using the whole corpora. The reference model M^{ref} comprises the lists of m top terms of each topic. Then, a fixed number of documents subsets are sampled (without replacement) from the corpora. For each documents subset G_i we generate the k-topic models M and compute the agreement between the reference model M^{ref} and M. With $\mathscr{M}$ we denote the set of k-topic models built from G_i. The computation of the agreement between two k-topic models is performed by: (1) building a square matrix J containing the average jaccard values of the k topics in M^{ref} (rows) and the k topics in M (columns); (2) computing the agreement score $agree(M^{ref}, M)$.

In particular, the average jaccard score for the k-th topic, with m top words, in M^{ref} and M is computed using the formula:

$$\overline{jacc}(M_k^{ref}, M_k) = \frac{1}{m} \sum_{l=0}^{m} \frac{|M_{k,l}^{ref} \cap M_{k,l}|}{|M_{k,l}^{ref} \cup M_{k,l}|}$$

The agreement score for the

$$agree(M^{ref}, M) = \frac{1}{k} \sum_{i=1}^{k} \max J_k$$

Finally the stability is computed as:

$$stability(k) = \frac{1}{|G|} \sum_{i=1}^{|G|} agree(M^{ref}, \mathscr{M}^i)$$

The best number of topics/clusters k is the one with highest stability score (mean agreement).

3.5 Case study: Learning user preferences and behaviour in open spaces

In this case study we present how user-space interaction data can be leveraged to learn tourists' behaviour in the scenario of visiting a cultural heritage centre.

3.5.1 Available data

The dataset we employ consists of 1663 users' POI-vist trajectories in the city of Florence (Italy) that have been reconstructed by employing data harvested from the Flickr photo sharing platform. The POI-visit trajectories are built by following the general example presented in Section 3.2. In particular, images in a Flickr photo album are tagged with information about the geographical coordinates and the shooting time, from these information a trajectory is constructed as follows: (1) geographical coordinates are used to represent the picture as a recognizable POI by fetching relevant content data from an external source; (2) the shooting time is used to order the identified POIs in such a way that we obtain a temporally ordered list of user's visited POI, i.e., the user itinerary. So, photo albums which elements fall within the geographical boundaries of the city of Florence[4] have been downloaded and sorted. Each photo is then matched with the Wikipedia pages whose geographical coordinates fall within the Florence area. The matching procedure is done by defining a circular area, with fixed

[4]https://www.openstreetmap.org/relation/42602#map=12/43.7716/11.3291

radius ($r = 100$ meters), centred in the photo coordinates and then by seeking for the closest Wikipedia content geo-localized in that area. In this case study we use as starting point the POI-visits trajectories dataset presented in [40]. We manually added to each POI-visit data content information about the POI itself by using expert knowledge extracted from the POI Wikipedia page. Since all the identified POIs are cultural attractions we decided to identify the following set of features to represent them: the POI category (e.g., monument), the historical period (i.e., century) and one historical person related to the POI. In the 532 POIs appearing in the trajectories we identified 13 different POI categories, 18 historical periods and 106 historical persons. With regard to the visit context of a POI-visits we leveraged the timestamp (the date) and the geographical coordinates of each POI-visit to query a weather service[5] to collect an hourly weather summary (e.g., cloudy), temperature (e.g., cold) and daytime (e.g., evening).

The trajectories/users ratio is 1.43 and the average trajectory length is 11.7 POI-visit.

3.5.2 Identification and inspection of like-behaving users

In order group like-behaving users in the dataset we apply the approach described in Section 3.4. So, we generated a text corpora by building a document-like representation for each trajectory. The terms in the corpora are the content and context features associated to the POI-visits. Then, we applied NMF and we identified 5 different trajectory clusters. In Table 3.1 we show the top-10 terms per cluster and the number of associated trajectories. Clusters are named with the first 5 English alphabet letters.

Table 3.1 Top 10 terms in the five topics extracted from the trajectory data set and number of trajectories assigned to each topic (cluster).

#Term	**Cluster A**	**Cluster B**	**Cluster C**	**Cluster D**	**Cluster E**
1	morning	hot	cloudy	warm	freezing
2	cold	afternoon	cold	cloudy	cloudy
3	square	century 16	church	century 14	afternoon
4	palace	palace	square	church	century 14
5	century 15	church	century 13	square	palace
6	century 13	square	palace	building	building
7	church	century 19	rain	palace	century 13
8	night	century 13	museum	ponte	church
9	dante	museo	brunelleschi	century 13	foggini
10	century 10	brunelleschi	tadda	century 19	century 19
#Traj.	368	339	341	297	153

[5]https://darksky.net

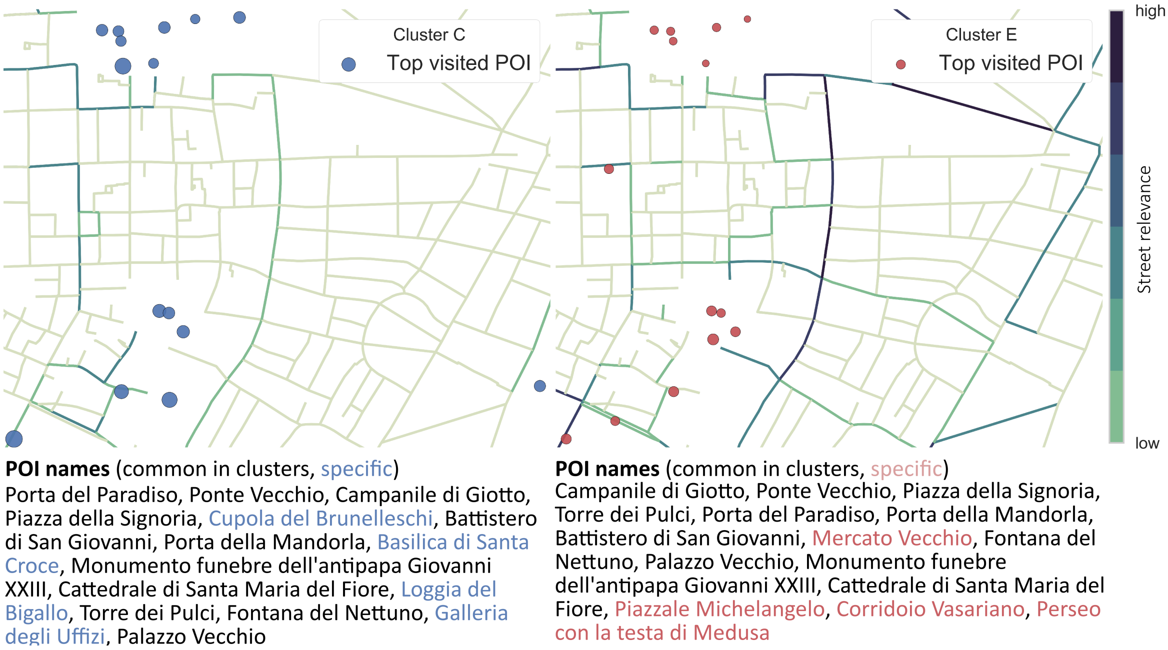

Fig. 3.1 Top-15 visited POIs and street relevance (heatmap) for two clusters

In the following, we exemplify the results of the clustering step by comparing two cluster examples (i.e., C and E); they have some similar features but a different number of trajectories. Figure 3.1 depicts the clusters' trajectories and 15 most popular POIs.

The POIs are depicted as circles with diameter proportional to the normalised POI popularity: the more popular the POI is in the cluster the larger the circle is. There is a large number of POIs present in both clusters, but they differ in terms of normalised visit frequency. In fact, in the cluster represented on the right (cluster E), POI circles are smaller because of a more uniform distribution of the visits among all the POIs in the cluster (i.e., not only the top-15 shown in the figure). An aspect that we see of particular interest is related to how users interacts with the surrounding environments. To show that, in Figure 3.1 we show how important are the streets of Florence for the clustered users/trajectories. The importance of the various streets in the clustered trajectories is determined by identifying the most representative trajectories in the clusters. These are the trajectories whose *tf-idf* vector representation is closer, in cosine similarity, to the cluster centroid, which is the average vector of all the *tf-idf* vector trajectory representations. The street importance is represented as shades of the colour bar on the right part of the figure; it has higher values (darker colour) in proximity to popular POIs and on the main streets connecting them.

At the bottom of Figure 3.1 the most popular POIs in the two clusters are listed. POI in black typeface are common to the two clusters, whereas coloured POIs are cluster specific.

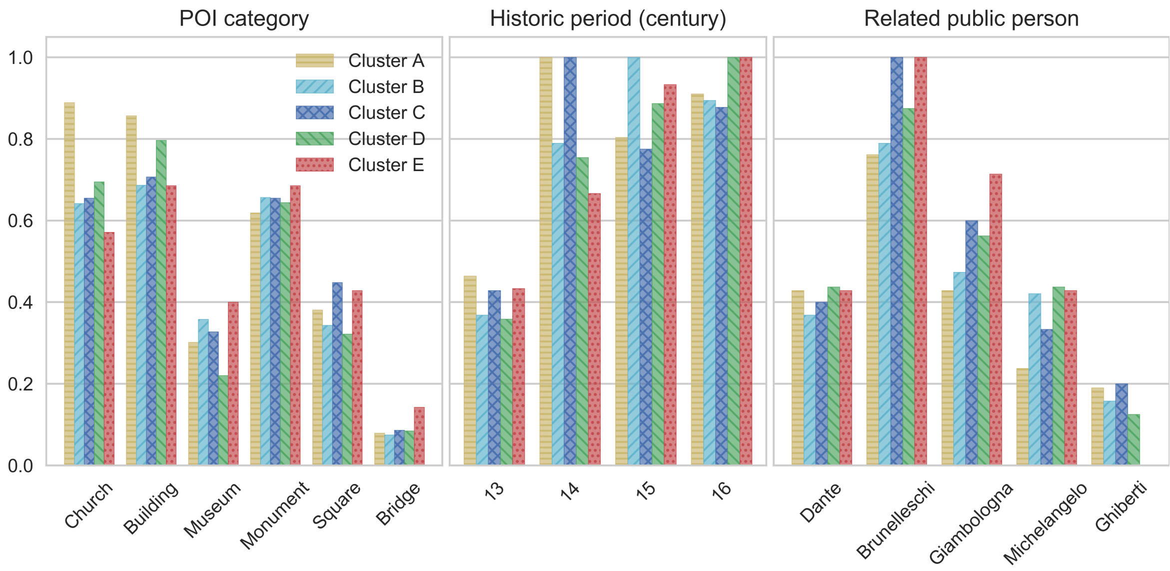

Fig. 3.2 Extract of POI features distribution per cluster.

11 POIs are common to these two clusters and 9 of them are common to all the clusters. They actually belong to the top-15 attractions according to popular travel portals[6].

We show additional clusters differences in Figure 3.2, where POI features per cluster are compared. This figure shows the probability of various features (POI category, historic period and related person features) in the five considered clusters.

Overall, we can see that the features variability in the clusters is not high. In fact, POIs are rather similar, i.e., they are mostly cultural POIs. It is reasonable to conjecture that if a more diverse assortment of POIs (e.g., leisure, restaurant, bar, etc.) were available then the clusters may have better discriminated alternative groups/types of tourists. Nevertheless, by looking at the specific POI descriptive features, one can notice interesting differences. For instance, POI categories like churches and buildings characterise mostly visits in clusters A and D, whereas to a lower extent trajectories in the other clusters (e.g., cluster E). Instead, cluster E is more representative of visits to bridges, squares and museums. Also POI historic period and POI related person features differentiate the clusters. For instance, cluster E is characterised by visits to POIs from the 15^{th} and 16^{th} centuries and artists from these times (i.e., Brunelleschi, Michelangelo and Giambologna). Other relations between historic period and related person can be identified in 13^{th} century and Dante (e.g., clusters A and C) as well as 13^{th} century and Ghiberti (e.g., cluster A). Carrying out this analysis with a domain expert, an art historian, could reveal more similarities and differences between the clusters.

[6] www.planetware.com/tourist-attractions-/florence-i-to-f.html
www.touropia.com/tourist-attractions-in-florence/
theculturetrip.com/europe/italy/articles/20-must-visit-attractions-in-florence-italy/

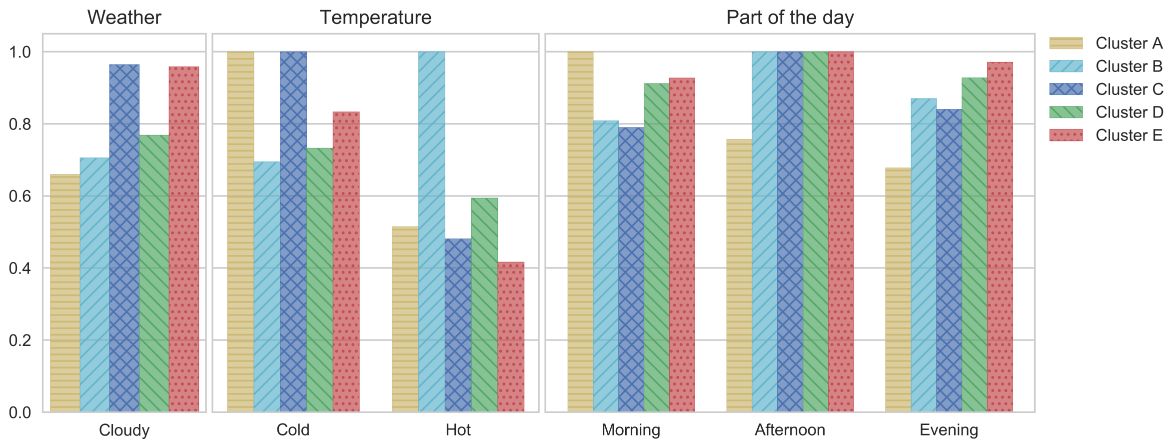

Fig. 3.3 Extract of context features distribution per cluster.

In Figure 3.3 we show the probabilities to observe certain context features in the clusters. For instance, by looking at the clusters C and E we can see that they mainly group visits during cloudy days (left). Considering instead the temperature (centre), the clusters capture other nuances of the visits. For instance, cluster A represents visits in cold days, whereas cluster C groups visits in warmer days. Interestingly, focusing on the part of the day (right), there are clusters that represent visits performed at different times. For instance, mornings and afternoons in cluster A, afternoons and evenings in clusters B and over the whole day cluster D.

By means of a χ^2 test of independence, it has been found that the frequency of POI category, historic period, related person and weather depend on the cluster (all significant with $p < 0.04$).

3.5.3 MDP modelling

We model each POI-visit trajectory in a cluster as the itinerary followed by a user, who is belonging to a group of like-minded users, that is taking decision as to optimize an (unknown) reward function common to all the users in the cluster. This problem is modelled as an MDP.

Let P be the set of POIs visited by the users and let $\phi(s)$ be the binary vector that represents for each POI the presence or absence of the following attributes: weather f_w, where $w \in \{$*clear, foggy, partly cloudy, mostly cloudy, rainy, windy*$\}$; temperature f_t, where $t \in \{$*freezing, cold, warm,hot*$\}$; daytime f_d, where $d \in \{$*morning, afternoon, evening, night*$\}$; POI category f_c, where $c \in \{$*church, ..., palace*$\}$; historic period f_h, where $h \in \{3^{rd}$ *century, ...,* 20^{th} *century*$\}$; related person f_r, where $r \in \{$*Brunellschi, ..., Vasari*$\}$. In total there are 151 Boolean features ($F = 151$), 137 representing the POI ($X = 137$) and 14 representing the context ($C = 14$).

We define the state space as $S = P \times C$ where a state s models the visit of a tourist at a specific POI in context.

In our problem a tourist can reach from a POI any other POI, therefore the set of actions is $A = P$. It is important to highlight that reaching a POI to visit next, i.e., performing an action, is a stochastic process: following action a to reach a next POI may lead to the desired place with different context conditions, e.g., at Battistero can be rainy or foggy.

We denote with ζ_u a user u trajectory, which is a temporally ordered list of states. 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}.

The transition model T is derived from the clustered trajectories. Since we are interested in learning long term reward, i.e., optimizing for the whole visit, we set $\gamma = 0.9$.

3.5.4 Tourist behaviour

In the second version of the thesis I will show the learnt behaviour models (per cluster) by showing why user's acted in a specific way.

Chapter 4

Harnessing user behaviour models for next-item recommendation

4.1 Recommendation strategies

- IRL-based
  - Q-BASE
  - Q-PREFERENCE
  - Q-POP
  - Q-POP PUSH
- Content-based

4.2 Case study: off-line performance of next-POI recommendations

…and some more

4.3 Case study: User evaluation of next-POI recommendations

…and some more

Chapter 5

Transferring the behaviour learnt in an area (city) to other areas (city)

5.1 Problem statement

Chapter 6

Exploiting off-line and on-line user behaviour for next-POI recommendations in a real application: the case of Wondervalley

Chapter 7

Discussion and Conclusion

References

[1] Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145.

[2] Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 217–253.

[3] Aiello, L. M., Schifanella, R., Quercia, D., and Prete, L. D. (2019). Large-scale and high-resolution analysis of food purchases and health outcomes. *EPJ Data Sci.*, 8(1):14:1–14:22.

[4] Atzori, L., Iera, A., and Morabito, G. (2011). SIoT: Giving a social structure to the internet of things. *IEEE Communications Letters*, 15(11):1193–1195.

[5] Babes-Vroman, M., Marivate, V., Subramanian, K., and Littman, M. (2011). Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning - ICML'11*, pages 897–904.

[6] Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. (2012). Context relevance assessment and exploitation in mobile recommender systems. In *Personal and Ubiquitous Computing*, volume 16, pages 507–526.

[7] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109 – 132.

[8] Braunhofer, M. and Ricci, F. (2017). Selective contextual information acquisition in travel recommender systems. *Information Technology and Tourism*, 17(1):5–29.

[9] Braunhofer, M., Ricci, F., Lamche, B., and Wörndl, W. (2015). A context-aware model for proactive recommender systems in the tourism domain. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, MobileHCI '15, pages 1070–1075, New York, NY, USA. ACM.

[10] Burke, R. (2007). *Hybrid Web Recommender Systems*, pages 377–408. Springer Berlin Heidelberg, Berlin, Heidelberg.

[11] Cantador, I., Fernández-Tobías, I., Berkovsky, S., and Cremonesi, P. (2015). Cross-domain recommender systems. In *Recommender Systems Handbook*, pages 919–959.

[12] Cavada, D., Elahi, M., Massimo, D., Maule, S., Not, E., Ricci, F., and Venturini, A. (2018). Tangible tourism with the internet of things. In *Information and Communication Technologies in Tourism 2018, ENTER 2018, Proceedings of the International Conference in Jönköping, Sweden, January 24-26, 2018.*, pages 349–361.

[13] Choudhury, M. D., Feldman, M., Amer-Yahia, S., Golbandi, N., Lempel, R., and Yu, C. (2010). Automatic construction of travel itineraries using social breadcrumbs. In *HT'10, Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Toronto, Ontario, Canada, June 13-16, 2010*, pages 35–44.

[14] Darden, M. (2017). Smoking, expectations, and health: A dynamic stochastic model of lifetime smoking behavior. *Journal of Political Economy*, 125(5):1465–1522.

[15] Ermon, S., Xue, Y., Toth, R., Dilkina, B., Bernstein, R., Damoulas, T., Clark, P., DeGloria, S., Mude, A., Barrett, C., and Gomes, C. P. (2015). Learning Large Scale Dynamic Discrete Choice Models of Spatio-Temporal Preferences with Application to Migratory Pastoralism in East Africa. pages 644–650.

[16] Felfernig, A., Friedrich, G., Jannach, D., and Zanker, M. (2015). *Constraint-Based Recommender Systems*, pages 161–190. Springer US, Boston, MA.

[17] Gede, M. and Kádár, B. (2019). Analysing tourism movements along the danube river based on geotagged flickr photography. *Proceedings of the ICA*, 2:37.

[18] Greene, D., O'Callaghan, D., and Cunningham, P. (2014). How many topics? Stability analysis for topic models. In *Machine Learning and Knowledge Discovery in Databases*, volume 8724 LNAI, pages 498–513.

[19] Gurbanov, T., Ricci, F., and Ploner, M. (2016). Modeling and predicting user actions in recommender systems. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, UMAP '16, pages 151–155, New York, NY, USA. ACM.

[20] Hariri, N., Mobasher, B., and Burke, R. (2012). Context-aware music recommendation based on latenttopic sequential patterns. In *Proceedings of the 6th ACM conference on Recommender systems - RecSys '12*, page 131.

[21] Hashemi, S. H. and Kamps, J. (2017). Where to go next?: Exploiting behavioral user models in smart environments. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, pages 50–58, New York, NY, USA. ACM.

[22] Hinton, G. E. and Sejnowski, T. J. (1999). Unsupervised learning : foundations of neural computation.

[23] Hristova, D., Aiello, L. M., and Quercia, D. (2018). The new urban success: How culture pays. *Frontiers in Physics*, 6:27.

[24] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative Filtering for Implicit Feedback. *IEEE International Conference on Data Mining*, pages 263–272.

[25] Jannach, D. and Lerche, L. (2017). Leveraging Multi-Dimensional User Models for Personalized Next-Track Music Recommendation. In *Proceedings of the Symposium on Applied Computing - SAC'17*, pages 1635–1642.

[26] Jia, X., Sun, F., Li, H., Cao, Y., and Zhang, X. (2017). Image multi-label annotation based on supervised nonnegative matrix factorization with new matching measurement. *Neurocomputing*, 219:518 – 525.

[27] Joseph, K., Tan, C. H., and Carley, K. M. (2012). Beyond "local", "categories" and "friends": clustering foursquare users with latent "topics". In *UbiComp*.

[28] Kahneman, D. and Smith, V. (2002). Foundations of behavioral and experimental economics. Nobel Prize in Economics documents 2002-1, Nobel Prize Committee.

[29] Kennan, J. and Walker, J. R. (2011). The Effect of Expected Income on Individual Migration Decisions. *Econometrica*, 79(1):211–251.

[30] Koren, Y. and Bell, R. (2015). *Advances in collaborative filtering*, pages 78–118. Springer US, Boston, MA.

[31] Kubitza, T., Voit, A., Weber, D., and Schmidt, A. (2016). An iot infrastructure for ubiquitous notifications in intelligent living environments. pages 1536–1541.

[32] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

[33] Li, S., Xu, L. D., and Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259.

[34] Li, Y., Hou, M., Liu, H., and Liu, Y. (2012). Towards a theoretical framework of strategic decision, supporting capability and information sharing under the context of internet of things. *Information Technology and Management*, 13(4):205–216.

[35] Lommatzsch, A., Kille, B., and Albayrak, S. (2017). Incorporating context and trends in news recommender systems. In *Proceedings of the International Conference on Web Intelligence*, WI '17, pages 1062–1068, New York, NY, USA. ACM.

[36] Lops, P., De Gemmis, M., and Semeraro, G. (2011). *Content-based recommender systems: State of the art and trends*, pages 73–105. Springer US, Boston, MA.

[37] Mahmood, T., Ricci, F., and Venturini, A. (2009). Improving Recommendation Effectiveness: Adapting a Dialogue Strategy in Online Travel Planning. *Information Technology & Tourism*, 11(4):285–302.

[38] Mobasher, B., H. Dao, T. Luo, and 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.*, pages 669–672.

[39] Moling, O., Baltrunas, L., and Ricci, F. (2012). Optimal radio channel recommendations with explicit and implicit feedback. In *Proceedings of the 6th ACM conference on Recommender systems - RecSys '12*, page 75.

[40] Muntean, C. I., Nardini, F. M., Silvestri, F., and Baraglia, R. (2015). On learning prediction models for tourists paths. *ACM Trans. Intell. Syst. Technol.*, 7(1):8:1–8:34.

[41] Nadai, M. D., Staiano, J., Larcher, R., Sebe, N., Quercia, D., and Lepri, B. (2016). The death and life of great italian cities: A mobile phone data perspective. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 413–423.

[42] Neu, G. and Szepesvári, C. (2009). Training parsers by inverse reinforcement learning. *Machine Learning*, 77(2-3):303–337.

[43] Newson, P. and Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009), November 4-6, Seattle, WA*, pages 336–343.

[44] Ng, A. and Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning - ICML '00*, pages 663–670.

[45] Ng, P. C., She, J., and Park, S. (2017). Notify-and-interact: A beacon-smartphone interaction for user engagement in galleries. In *2017 IEEE International Conference on Multimedia and Expo, ICME 2017, Hong Kong, China, July 10-14, 2017*, pages 1069–1074.

[46] Niyato, D., Hossain, E., and Camorlinga, S. (2009). Remote patient monitoring service using heterogeneous wireless access networks: Architecture and optimization. *IEEE Journal on Selected Areas in Communications*, 27(4):412–423.

[47] Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*, volume 1 of *MIT Press Books*. The MIT Press.

[48] Palumbo, E., Rizzo, G., and Baralis, E. (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*, pages 1–8.

[49] Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., and Yang, Q. (2008). One-class collaborative filtering. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 502–511.

[50] Petrelli, D., Ciolfi, L., van Dijk, D., Hornecker, E., Not, E., and Schmidt, A. (2013). Integrating material and digital: A new way for cultural heritage. *interactions*, 20(4):58–63.

[51] Poikela, M., Schmidt, R., Wechsung, I., and Möller, S. (2014). Locate!-When do Users Disclose Location? In *Workshop on Privacy Personas and Segmentation (PPS) at the Tenth Symposium On Usable Privacy and Security (SOUPS).*

[52] Quercia, D., Aiello, L. M., and Schifanella, R. (2018). Diversity of indoor activities and economic development of neighborhoods. *PLOS ONE*, 13(6):1–18.

[53] Ramachandran, D. and Amir, E. (2007). Bayesian inverse reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence 2007 - IJCAI'07*, pages 2586–2591.

[54] Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. (2006). Maximum margin planning. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 729–736.

[55] Ren, B., Pueyo, L., Zhu, G. B., Debes, J., and Duchêne, G. (2018). Non-negative matrix factorization: Robust extraction of extended structures. *The Astrophysical Journal*, 852(2):104.

[56] Ricci, F., Rokach, L., and Shapira, B. (2015). Recommender systems: Introduction and challenges. In Ricci, F., Rokach, L., and Shapira, B., editors, *Recommender Systems Handbook*, pages 1–34.

[57] Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition.

[58] Shaer, O. and Hornecker, E. (2009). Tangible user interfaces: Past, present and future directions. *Foundations and Trends in Human-Computer Interaction*, 3(1-2):1–137.

[59] Shani, G., Heckerman, D., and Brafman, R. I. (2005). An mdp-based recommender system. *Journal of Machine Learning Research*, pages 1265–1295.

[60] Silva, T. H., de Melo, P. O. S. V., Almeida, J. M., Musolesi, M., and Loureiro, A. A. F. (2014). You are what you eat (and drink): Identifying cultural boundaries by analyzing food and drink habits in foursquare. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.*

[61] Silva, T. H., Viana, A. C., Benevenuto, F., Villas, L., Salles, J. F. S., Loureiro, A. A. F., and Quercia, D. (2019). Urban computing leveraging location-based social network data: A survey. *ACM Comput. Surv.*, 52(1):17:1–17:39.

[62] Sutton, R. S. and Barto, A. G. (2014). *Reinforcement Learning: An Introduction (Second edition, in progress)*. The MIT Press.

[63] Suzuki, S. (2018). Comparative analysis of human movement prediction: Space syntax and inverse reinforcement learning. *ArXiv*, abs/1801.00464.

[64] Taslaman, L. and Nilsson, B. (2012). A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data. In *PloS one*.

[65] Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 5th ACM conference on Recommender systems - RecSys '11*, page 109.

[66] Yin, M., Sheehan, M., Feygin, S., Paiement, J., and Pozdnoukhov, A. (2018). A generative model of urban activities from cellular data. *IEEE Trans. Intelligent Transportation Systems*, 19(6):1682–1696.

[67] Zhang, J.-D., Chow, C.-Y., and Li, Y. (2014). Lore: Exploiting sequential influence for location recommendations. In *Proceedings of the 22Nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '14, pages 103–112, New York, NY, USA. ACM.

[68] Zheng, Y., Capra, L., Wolfson, O., and Yang, H. (2014). Urban computing: Concepts, methodologies, and applications. *ACM TIST*, 5(3):38:1–38:55.

[69] Zheng, Y., Xie, X., and Ma, W.-Y. (2010). Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data(base) Engineering Bulletin*.

[70] Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - AAAI'08*, pages 1433–1438.