Expert Systems With Applications 69 (2017) 1–9

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Planning for tourism routes using social networks

Isabel Cenamor*, Tomás de la Rosa, Sergio Núñez, Daniel Borrajo

*Computer Science Department, Universidad Carlos III de Madrid, Leganés (Madrid), Spain*

ARTICLE INFO

*Article history:*
Received 6 April 2015
Revised 12 October 2016
Accepted 13 October 2016
Available online 19 October 2016

*Keywords:*
Tourism routes
Automated planning
Recommender systems

ABSTRACT

Traveling recommendation systems have become very popular applications for organizing and planning tourist trips. Among other challenges, these applications are faced with the task of maintaining updated information about popular tourist destinations, as well as providing useful tourist guides that meet the users preferences. In this work we present the PlanTour, a system that creates personalized tourist plans using the human-generated information gathered from the minube[1] traveling social network. The system follows an automated planning approach to generate a multiple-day plan with the most relevant points of interest of the city/region being visited. Particularly, the system collects information of users and points of interest from minube, groups these points with clustering techniques to split the problem into per-day sub-problems. Then, it uses an off-the-shelf domain-independent automated planner that finds good quality tourist plans. Unlike other tourist recommender systems, the PlanTour planner is able to organize relevant points of interest taking into account user's expected drives, and user scores from a real social network. The paper also highlights how to use human provided recommendations to guide the search for solutions of combinatorial tasks. The resulting intelligent system opens new possibilities of combining human-generated knowledge with efficient automated techniques when solving hard computational tasks. From an engineering perspective we advocate for the use of declarative representations of problem solving tasks that have been shown to improve modeling and maintenance of intelligent systems.



## 1. Introduction

Tourism is an important social, cultural and economic phenomenon that includes the movement of millions of people around the world with a big impact on the economy of many countries. Therefore, the generation of tourism-related tools can have a huge impact in society. Traveling recommendation systems have become very popular applications for organizing and planning tourist trips (Berka & Plößnig, 2004; Castillo et al., 2008; Moreno, Valls, Isern, Marin, & Borràs, 2013; Vansteenwegen, Souffriau, Berghe, & Oudheusden, 2011). One of the main bottlenecks of this type of systems consists of the initial population and later maintenance of the information about Points Of Interest (POIs), user ratings, and connection with geographic systems. However, in recent years we have seen the emergence of new social network platforms where users can easily and are willing to update that kind of information (e.g. TripAdvisor[2] or minube[3]). Also, the extensive use of tourist mobile applications allows users to request real time information about the schedules, guides or plans that fulfill their preferences (Rodriguez-Sanchez, Martinez-Romo, Borromeo, & Hernandez-Tamames, 2013). Data might come from different services, so middle layers should be developed, as wrappers and crawlers that obtain and integrate available data.

From a crowdsourcing perspective (Manikonda, Chakraborti, De, Talamadupula, & Kambhampati, 2014), users of traveling social networks do not receive an explicit call for supplying relevant tourist information or composing a plan. Instead, they are encouraged to share their experience of past trips and give recommendations to everyone. Therefore, users are helping to acquire personalized relevant information using a collaborative filtering mechanism (Lucas et al., 2012). Collaborative filtering provides a subset of recommendations on what to visit, where to sleep or where to eat. Additionally, the network structure facilitates the acquisition of personalized information related to user's contacts, which can greatly help on weighting the recommendations by closeness to the user.

As in other application areas, once the data collection and maintenance of information has a reasonable solution, people look at applications that build on top of that data. One such type of added-value applications on the tourism sector is based on automatically generating tourist plans. Currently, there are some

* Corresponding author. Fax: +34916249129.
*E-mail addresses:* icenamor@inf.uc3m.es (I. Cenamor), trosa@inf.uc3m.es (T. de la Rosa), sncovarr@pa.uc3m.es (S. Núñez), dborrajo@ia.uc3m.es (D. Borrajo).

[1] www.minube.com
[2] www.tripadvisor.com
[3] www.minube.com

http://dx.doi.org/10.1016/j.eswa.2016.10.030
0957-4174/

platforms that provide related services. For instance, Tripomatic[4] is a powerful tool for travel planning, but requires the user to select places to visit and manually set up the plan. As another example, CityTripPlanner[5] is able to automatically generate tourist plans, but it does not suggest places to eat in a reasonable way according to user pace, hunger and restaurant timetables.

In this paper we present PlanTour, a new tool that uses an automated planning approach to generate tourist plans. This planning system was built for the ondroad project, a framework for the management and planning of digital contents and services provided for bus travelers of the ALSA[6] company. Within this project, PlanTour is the sub-system in charge of building the tourist plans for users visiting a particular city or region.

In terms of planning applications, the main contributions of this work can be summarized as:

- The automatic composition of the initial state and goals using information from a social network. This partially tackles the problem of the slow start of similar systems, which have to wait until a sufficient amount of data is collected to run properly.
- The modeling of user drives as part of the planning process, to obtain more realistic plans in terms of deliberative reasoning. Specifically, suggested restaurants are smoothly integrated in tourist plans just when it is expected that the user is hungry based on her preferences.
- Modeling the problem of recommending tourist POIs as an oversubscription planning task, given that the available alternatives (visiting POIs) are many more than the ones a tourist can carry out with her time and budget.
- Successful application of compiling away soft goals using the approach by Keyder and Geffner (2009) to solve the oversubscription planning task.
- The domain-dependent problem decomposition using a clustering algorithm. Resulting sub-problems match the problem of finding a plan for a single day where candidate POIs are geographically close.

The following sections describe the problem formulation, the architecture with all its components, the representation of the domain, the experimental results, the related work, and finally, conclusions and future work.

## 2. Problem formulation

The generation of personalized tourist plans has been previously proposed as a *Tourist Trip Design Problem* (TTDP) (Gavalas, Konstantopoulos, Mastakas, & Pantziou, 2014). This generic class of problems comprises a set of candidate POIs together with their associated attributes (i.e., type, location, timetable, etc.), travel time between POIs, user-dependent functions relative to POIs (i.e., satisfaction, expected duration, etc.), the trip time-span and the daily time limit. A quality solution to a TTDP is expected to suggest a daily plan that respects the constraints imposed by POIs properties while maximizing the user utility. Modeling a concrete TTDP will depend on the available tourist data and the user inputs. In PlanTour the TTDP is modeled with the following elements:

- A fully-connected graph $(V, E)$ where every node $v \in V$ represents a POI and every edge $e \in E$ represents a path between two different locations. A function $d_t : E \rightarrow \mathbb{R}^+$ represents the time it takes to traverse an edge. The possible routes between the locations do not always use the same transportation means. Depending on the distance between two locations, the edge can be traversed either walking or by car. Thus, the values of the time function $d_t$ are computed considering these two options.
- A tuple of properties associated to every POI in $V$. Thus, each property is denoted by its function name (e.g $type(v)$, $duration(v)$). These properties are summarized in Table 1.
- A tuple of properties associated to the user for encoding her constraints and preferences. For instance, the time-span *[init_trip,end_trip]* for the whole visit. Table 2 shows the list of user properties. Given that our plans also include eating actions, we define $V_r \subset V$ as a special set of POIs representing the places where the user can eat. Plans should consider the number of times the user wants to eat in a day (*times_eat*). Also, the selection of restaurants should respect the minimum ($h_{\min}$) and maximum ($h_{\max}$) time that the user considers reasonable to wait until eating again. These constraints forbid the concatenation of two POIs belonging to $V_r$, but force to include some of them when the user is expected to be hungry.

**Table 1**
Properties included in the TTDP formulation to characterize POIs.

| Property | Values |
|---|---|
| Type | Nature, building, sport, eating, events, ... |
| Subtype | Subtype within a type (e.g. a museum is a subtype of building) |
| Score | The POI *total_score* taken from minube (POI popularity) |
| Timetable | The opening (*t_open*) and closing time (*t_close*) |
| Duration | Estimated time to visit in function of the subtype ($t_p$) |
| Location | The POI geo-coordinates (latitude and longitude) |

**Table 2**
Properties of users that can be specified from his/her preferences.

| Property | Description |
|---|---|
| Location | Where the user is |
| *[init_trip,end_trip]* | The trip time-span |
| $h_{\min}$ | Minimum time to pass to recommend a place to eat |
| $h_{\max}$ | Maximum time to pass to recommend a place to eat |
| *times_eat* | Maximum number of times the user eats on a day |
| *init_time* | The first time the user is available in the day |
| *end_time* | The last time the user is available in the day. When this Time is due, we assume the user is tired to do more things |

A solution to this TTDP is a sequence of time-stamped actions, which includes tourism activities and the necessary travels to perform these activities. This plan should satisfy the time-span constraint for the whole visit and the daily time constraints, where the user wants to perform the activities.

## 3. System architecture

The PlanTour architecture is composed of three main subservices (see Fig. 1). The Tourist Plan Manager (TPM) receives the inputs for the PlanTour planner. The inputs of PlanTour are: the city or region the user is going to visit; when s/he is going to be available (time to arrive and leave the place); and possibly, some constraints and preferences. Users are not required to input a complete list of their constraints and preferences. Therefore, we provide default values for these properties. The default values are: *[init_time, end_time]* = [10:00 AM, 8:00 PM], the time interval that the user can use to perform an activity, *times_eat = 2*, only two POIs from $V_r$ will be recommended in a day; and $h_{\min} = 2$ and $h_{\min} = 5$, so that at least two and at most five hours should pass before recommending a place from $V_r$ again.

When a plan is requested by the user, the TPM retrieves the needed information from the minube social network. This information is pre-processed to create planning problems in the standard declarative language used by the automated planning

[4] www.tripomatic.com
[5] citytripplanner.com
[6] http://www.alsa.es/en