Exploiting Item Dependencies to Improve Tourist Trip Recommendations

Daniel Herzog
Department of Informatics
Technical University of Munich
Boltzmannstr. 3, 85748 Garching, Germany
herzogd@in.tum.de

Wolfgang Wörndl
Department of Informatics
Technical University of Munich
Boltzmannstr. 3, 85748 Garching, Germany
woerndl@in.tum.de

ABSTRACT

Combining multiple points of interest (POIs) to attractive and reasonable tourist trips is a challenge in the field of Recommender Systems (RSs). Even if a user likes going to restaurants, a trip composed of too many restaurants will not be appreciated. In this position paper, we present our ideas how to improve tourist trip recommendations by focusing more on user satisfaction. We introduce the concept of item dependencies describing how POIs influence the value of other POIs in the same trip when recommending tourist trips. Besides background information and related work in the field of tourist trip recommendations, we present ideas to iteratively learn dependencies between items and to integrate them into the recommendation process.

CCS Concepts

•**Information systems** → **Recommender systems;** •**Human-centered computing** → *Human computer interaction (HCI);*

Keywords

Item Dependencies, Sequential Recommendation, Tourist Trip Design Problem, POI, User Interface

1. INTRODUCTION AND MOTIVATION

Optimizing sequences of recommendations is an ongoing challenge in the research of Recommender Systems (RSs) [13]. One example of sequential recommendations are tourist trips composed of multiple points of interest (POIs) such as restaurants, museums or monuments. Finding the right combination of POIs for a tourist trip is a complex task. Combining the highest rated POIs into a sequence does not guarantee the highest possible user satisfaction when one POI has a negative influence on another POI or the trip itself. For example, a person who likes going to restaurants will most likely prefer daily trips including one or two restaurants but every additional restaurant may be less appealing. On the other side, a craft market might be more appreciated after visiting a related folk museum [19]. The examples show that the total value of a trip is not the sum of the predicted ratings of the POIs. Instead, the value of a POI for a user is influenced by other POIs in the same trip. We call this influence item dependencies. Such item dependencies can follow a general pattern (e.g., limiting restaurants in a trip to a reasonable number) but usually differ between users because of personal preferences.

In order to integrate item dependencies into the recommendation process, the user's preferences and the relevant item dependencies for the user have to be collected. Advanced user interfaces and interaction options help to achieve this goal. Thus, we want to tackle the described problem from two perspectives: recommendation algorithms and the user's perspective. We define the following two research questions:

RQ 1 How can existing algorithms be extended to consider item dependencies when recommending POI sequences?

RQ 2 How can user interfaces support the users in providing feedback on mobile devices with regard to appreciated combinations of POIs?

In order to find answers to these research questions, we will develop novel algorithms, implement them in a real working RS and evaluate their performance in large user studies. In this position paper, we start our research by presenting background information and related work. We introduce item dependencies in tourist trips and suggest a framework for sequential POI recommendations with the focus on finding the best combinations of POIs. In the end, we give an outlook on experiments we want to conduct to evaluate our work and we provide a short conclusion.

2. BACKGROUND AND RELATED WORK

In this section, we provide an introduction to the topic of tourist trip recommendations. We briefly summarize important related work in this field and introduce the extension of item dependencies.

2.1 Related Tourist Trip Design Problems

The problem of combining POIs to attractive and reasonable routes is called the Tourist Trip Design Problem (TTDP) [19]. In its simplest specification, the TTDP is identical to the Orienteering Problem (OP): every location which can be visited has a value but a time budget and the known travel time between the points restricts the number

RecTour 2016 - Workshop on Recommenders in Tourism held in conjunction with the 10th ACM Conference on Recommender Systems (RecSys), September 15, 2016, Boston, MA, USA.

of possible routes [15]. The OP aims to find a route which includes some of the points to maximize the overall value for the traveler while not exceeding the time budget.

Over the past years, different extensions of the OP have been researched. The team orienteering problem (TOP) aims at finding multiple routes at the same time while avoiding overlaps [4]. In the (T)OP with time windows (TOPTW), each location can only be visited within a defined time window (e.g., the opening hours of that POI) [18]. Further variants allow the integration of inter-modal transportation into the trip planning [6] or add multiple constraints [14].

A few variants of the OP pursue similar goals to our work. Little attention has been given to the Generalized Orienteering Problem (GOP) which can be applied to, for example, reduce the value of a trip if it contains many equal attractions. The main difference between the OP and the GOP is that every node in the GOP comes with a set of values representing multiple goals of the visitor [10]. Other variants close to our problem are the OP with variable profits (OPVP) [5], the TOP with decreasing profits (DPTOP) [1] and the Clustered OP (COP) [2]. The OPVP assumes that the node values depend on a number of discrete passes or the time spent at the node. In the DPTOP the profit of each node decreases with time and in the COP the score of a node can only be gained if all nodes of a group of nodes are part of the path.

Extensive overviews of existing algorithms and heuristics solving the described problems are presented by Vansteenwegen et al. [17], Gavalas et al. [9] and Gunawan et al. [11]. So far, no existing work considers individual dependencies between POIs, e.g., the influence of a restaurant on another POI. In our work, we want to develop heuristics that maximize the user satisfaction by incorporating item dependencies and that can be used for practical applications.

2.2 Existing Tourist Trip Applications

Some applications recommending sequences of items exist but only a few working prototypes recommend tourist trips. Vansteenwegen et al. developed the City Trip Planner, a web application that recommends trips for a requested number of days [16]. It respects limitations like opening hours and can include a lunch break into the trip. An updated version is available at www.citytripplanner.com. A similar application for multi-day tourist trips is DailyTRIP [8]. Wörndl and Hefele [21] developed a web application for finding city trips. It uses Foursquare to predict POI ratings for the user and extends Dijkstra's algorithm to generate routes. Garcia et al. developed a desktop and mobile prototype for recommending trips in San Sebastián [7]. mTrip (www.mtrip.com/en/travel-guide/) is a mobile tourist guide available for Android and iOS. Some of these applications allow basic customization after a trip has been recommended, e.g., removing single POIs or use more iterative dialogues between the user and the system to find travel packages [20] [12]. None of them provides advanced user interfaces to learn and consider individual dependencies between POIs, which is an important task when improving the selection of items in a sequential RS.

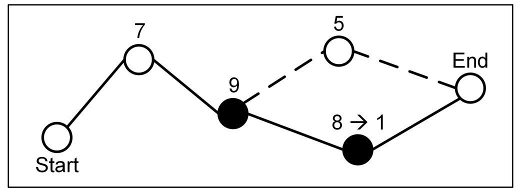

Figure 1: Item dependencies in a tourist trip

2.3 Item Dependencies in Tourist Trip Recommendations

In most of the OP variants, a location is a node with a fix value. As POIs come with certain characteristics (e.g., the POI type), we claim that the attractiveness of a tourist trip recommendation can be increased if these values are flexible and dependent on the presence or absence of other POIs in the same trip.

Figure 1 shows how considering item dependencies changes the trip generation process. In this example, the black points represent restaurants, the white points POIs of other categories. The predicted ratings are in a range from 1 (lowest value) to 10 (highest value). Assuming that a user does not have the time to visit all POIs, the route of the solid line could be recommended. However, two restaurants in a trip with three POIs might not be appreciated by the user. Thus, the rating of the second restaurant perceived by the user is actually lower than the prediction (1 instead of 8). Algorithms incorporating item dependencies would therefore change the trip by the dashed line to generate a more pleasant route (assuming that including the new POI does not have any negative influence on the other POIs of the trip).

The existing TTDP applications presented in Section 2.2 generate feasible routes but they do not consider the described dependencies between POIs. This is an important, open task to improve the quality of recommended tourist trips [21].

3. PROPOSED SOLUTION AND NEXT RESEARCH STEPS

To tackle the described problem, we have to develop novel algorithms considering dependencies between POIs. Furthermore, a RS has to provide user interfaces that allow to learn user preferences and item dependencies and to provide feedback on recommendations while minimizing user effort.

3.1 Extending Existing TTDP Algorithms

We focus on trip recommendations from a user perspective and for practical applications. Hence, we will mainly develop and improve heuristics instead of exact algorithms to ensure a feasible runtime.

Greedy algorithms choose the locally optimal choice at each step of the trip generation. One example is Dijkstra's algorithm which already has been used to recommend tourist trips [21]. Such an algorithm can be adapted to use flexible values that change depending on the already visited nodes of the graph. Other approaches solving the OP start with finding a path using a greedy algorithm and then update the path in an iterative manner, i.e., removing or replacing single nodes of the generated path [4]. This is another promising solution for incorporating item dependencies. After a first path has been found, single POIs can be replaced or removed

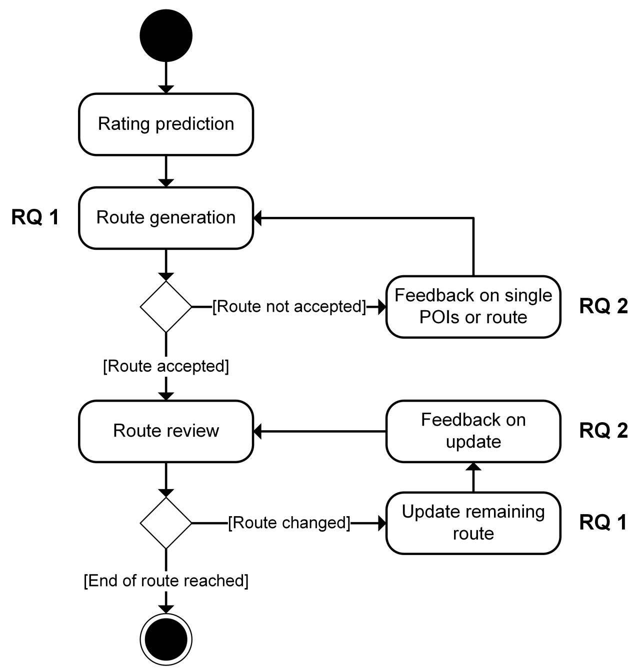

Figure 2: Activity diagram of our framework for generating and updating a tourist trip

if this has a positive effect on other POIs or the trip, as presented in Figure 1. Another idea is to extend a tabu search heuristic which already has been applied for more complex OP variants [14].

3.2 Creating Routes and Learning Item Dependencies

User preferences and relevant item dependencies have to be elicited to improve the outcome of the presented algorithms. One goal is to reduce user interaction especially when the user is moving or already on a trip.

We suggest a conversational recommendation approach. The idea is to provide dialogues to iteratively create and update the recommendations and to use the user's feedback to learn relevant item dependencies. The key activities of our framework are illustrated in Figure 2. After predicting ratings for all POIs that come into consideration for recommendation, two iterative processes generate POI sequences and update the recommendation if the user's plans change. The key activities are explained in detail in the following. The annotations in Figure 2 show which activities aim at solving the first (RQ 1) and which the second (RQ 2) research question.

The framework is composed of three main phases. In the *rating prediction* phase, established recommendation techniques are applied to predict ratings. These ratings represent the value of the POI for the user regardless of other POIs. This rating should consider the context of the recommendation to improve the prediction. For example, an outdoor POI should receive a lower rating when the weather is bad. In the next phase, *route generation*, the RS creates the first route including some of the rated POIs. Therefore, one of the algorithms introduced in Section 3.1 is applied. In contrast to single-shot recommendations, our framework generates routes in an iterative manner. For example, the user can be presented with two or more alternatives for concrete POI recommendations and can indicate her or his preferences for one POI over the others. Other options are suggestions for adding or removing POIs. While some dependencies are universal (e.g., no need for two restaurants in a row), these interactions support the RS in learning further combinations of POIs the user appreciates or rejects. Nevertheless, the user should not be overwhelmed with interactions. This is why implicit feedback plays an important role in our research. If, for example, a user spends a lot of time at a POI, it is likely that the user is interested in similar POIs. After each feedback phase, the RS suggests an optimized sequence based on the user's feedback. Finally, in the *route review* phase, the RS observes the user's progress and updates the rest of the current route when the user's plans change spontaneously. For example, when the user spends more time at a POI, visits additional POIs or skips suggested steps of the trip, the trip should be updated accordingly [19]. Again, interfaces allow the user to select her or his preferences if, for example, another POI should be added to the trip. The challenge is to update the route while considering the already visited POIs and their item dependencies. Furthermore, the system has to inform the user if a previously chosen POI cannot be visited during the trip anymore.

3.3 Evaluations and User Studies

In this section, we briefly want to outline our planned experiments and user studies for evaluating our work. This evaluation will be split in two parts: evaluating the recommendation algorithms and user studies for the developed user interfaces. In the end, a bigger, comprehensive study will be conducted to evaluate the RS as a whole.

A big selection of benchmark instances for the OP and its variants exist [17] [11]. However, our goal is not to find exact solutions for the OP with item dependencies. Instead, our focus are practical applications. This is why we tackle the problem with heuristics that provide satisfying solution in a feasible time. Another problem is that our approaches for solving the OP with item dependencies can not be compared to the benchmark instances of other variants. In our problem, the value of a node is flexible and depending on other nodes in the same path. Hence, the maximum total value of the trip can differ significantly. To tackle the described challenges, we will develop different algorithms considering item dependencies. Like this, we can compare the algorithms with each other and identify the most promising approaches. For a comparison with algorithms solving the TTDP without item dependencies, we will conduct user studies aiming at measuring the user satisfaction. We will present tourist trips created by different algorithms and let the user evaluate the quality of the trip and their satisfaction.

The second pillar of our experiments are user studies to measure the usability of the interfaces that support the user to create and improve tourist trips and to learn personal item dependencies. These interfaces will be developed in a iterative, user-centered approach. We will start with observations and interviews to elicit user requirements. Paper prototypes will allow us to evaluate the usability of our drafts before the actual implementation takes place. Different versions can be compared in A/B testing. The user feedback will be implemented in further developments of a functional prototype. To measure usability, established questionnaires like the System Usability Scale (SUS) can be used [3]. This questionnaire consists of ten questions providing a global view of subjective assessments of usability. Based on the responses, a SUS score can be calculated to measure usability and to compare different systems.

In the end, the developed interfaces will be integrated into a working application which will be evaluated in lab and field studies with real users.

4. CONCLUSION

In this paper, we targeted the issue of item dependencies in tourist trips. We presented a framework that can be used to iteratively generate and improve recommendations. The framework represents the starting point of our research in the field of sequential recommendations. The goal is to use it for the development of a real working mobile RS. Hence, our next step is to examine which existing TTDP algorithms can be extended to consider the influence of POIs on other POIs in a tourist trip. As there are no existing solutions considering dependencies, we have to develop multiple algorithms and compare them with regard to quality of the trips, a feasible runtime and user satisfaction.

The second key aspect of future work is the development and evaluation of interfaces facilitating the creation of pleasant sequences. They should allow the user to express her or his travel preferences and the application to learn relevant dependencies between POIs. When the user's plans change spontaneously, dialogues can support the modification of the trip. These dialogues must not be too distracting or annoying, especially when the user is moving. Thus, implicit feedback plays an important role.

The expected outcome of our research is a sequential RS that outperforms previous solutions with regard to attractiveness of the trips and usability of the system. We want to evaluate our algorithms and the conversational RS in large user studies in a realistic environment. This is why we will develop a mobile application for recommending POI sequences.

5. REFERENCES

[1] H. M. Afsar and N. Labadie. Team orienteering problem with decreasing profits. *Electronic Notes in Discrete Mathematics*, 41:285 – 293, 2013.

[2] E. Angelelli, C. Archetti, and M. Vindigni. The clustered orienteering problem. *European Journal of Operational Research*, 238(2):404 – 414, 2014.

[3] J. Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, pages 189–194, 1996.

[4] I.-M. Chao, B. L. Golden, and E. A. Wasil. The team orienteering problem. *European journal of operational research*, 88(3):464–474, 1996.

[5] G. Erdogan and G. Laporte. The orienteering problem with variable profits. *Networks*, 61(2):104–116, March 2013.

[6] F. V. Fomin and A. Lingas. Approximation algorithms for time-dependent orienteering. *Information Processing Letters*, 83(2):57–62, 2002.

[7] A. Garcia, O. Arbelaitz, M. T. Linaza, P. Vansteenwegen, and W. Souffriau. Personalized tourist route generation. In *Proceedings of the 10th International Conference on Current Trends in Web Engineering*, ICWE'10, pages 486–497, Berlin, Heidelberg, 2010. Springer-Verlag.

[8] D. Gavalas, M. Kenteris, C. Konstantopoulos, and G. Pantziou. Web application for recommending personalised mobile tourist routes. *Software, IET*, 6(4):313–322, 2012.

[9] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3):291–328, June 2014.

[10] Z. W. Geem, C.-L. Tseng, and Y. Park. Harmony search for generalized orienteering problem: best touring in china. In L. Wang, K. Chen, and Y. S. Ong, editors, *Advances in natural computation*, pages 741–750. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[11] A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 2016.

[12] T. Mahmood, F. Ricci, and A. Venturini. Improving recommendation effectiveness: Adapting a dialogue strategy in online travel planning. *Information Technology & Tourism*, 11(4):285–302, 2009.

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

[14] K. Sylejmani, J. Dorn, and N. Musliu. A tabu search approach for multi constrained team orienteering problem and its application in touristic trip planning. In *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*, pages 300–305, Dec 2012.

[15] T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, pages 797–809, 1984.

[16] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden. The city trip planner. *Expert Syst. Appl.*, 38(6):6540–6546, June 2011.

[17] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

[18] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden. Iterated local search for the team orienteering problem with time windows. *Comput. Oper. Res.*, 36(12):3281–3290, Dec. 2009.

[19] P. Vansteenwegen and D. Van Oudheusden. The mobile tourist guide: an or opportunity. *OR Insight*, 20(3):21–27, 2007.

[20] A. Venturini and F. Ricci. Applying trip@dvice recommendation technology to www.visiteurope.com. In *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, pages 607–611, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.

[21] W. Wörndl and A. Hefele. Generating paths through discovered places-of-interests for city trip planning. In *Information and Communication Technologies in Tourism 2016*, pages 441–453. Springer, 2016.