RecTour 2018

Workshop on Recommenders in Tourism

Vancouver, Canada, October 7th, 2018

Proceedings

Edited by

Julia Neidhardt, Wolfgang Wörndl,
Tsvi Kuflik and Markus Zanker

Co-located with the 12th ACM Conference on Recommender Systems (RecSys 2018)

The ACM Conference Series on
Recommender Systems

Copyright and Bibliographical Information

This volume is published by Julia Neidhardt, Wolfgang Wörndl, Tsvi Kuflik and Markus Zanker.

Published online at http://ceur-ws.org/Vol-2222/.

Proceedings of the Workshop on Recommenders in Tourism (RecTour 2018), held in conjunction with the 12th ACM Conference on Recommender Systems (RecSys 2018), October 2nd - 7th, 2018, Vancouver, Canada, https://recsys.acm.org/recsys18/.

Julia Neidhardt, Wolfgang Wörndl, Tsvi Kuflik and Markus Zanker (editors).

Further information about the workshop can be found at: http://www.ec.tuwien.ac.at/rectour2018/

Preface

This volume contains the contributions presented at the Workshop on Recommenders in Tourism (RecTour), hold in conjunction with the 12th ACM Conference on Recommender System (RecSys 2018), in Vancouver, Canada. The proceedings are also published online by CEUR Workshop Proceedings at http://ceur-ws.org/Vol-2222/.

RecTour 2018 focuses on a variety of challenges specific to recommender systems in the tourism domain. This domain offers considerably more complicated scenarios than matching travelers with the presumably best items. Planning a vacation usually involves searching for interconnected and dependent product bundles, such as means of transportation, accommodations, attractions, and activities, with limited availabilities and contextual aspects (e.g., spatio-temporal context, social context, activity sequence, and environment) with a major impact. In addition, travel related products are emotionally “loaded” and thus largely experiential in nature; therefore, decision taking is often not solely based on rational or objective criteria. Therefore, information provisioning at the right time about destinations, accommodations and various further services and possible activities is challenging. Additionally, and in contrast to many other recommendation domains, information providers are usually small and medium sized enterprises (SMEs) that many times do not possess the capacity to implement basic recommender systems. Moreover, there is no single, standard format to house information which might be included in these systems. Last, much of the tourism experience is co-produced, i.e., it occurs during the consumption of the product and interaction with the provider. Therefore, the context of the recommendation is extremely important. Thus given this diversity, building effective recommender systems within the tourism domain is extremely challenging. The rapid development of information and communication technologies (ICT) in general and the web in particular has transformed the tourism domain whereby most travelers rely little on travel agents or agencies. Indeed, recent studies indicate that travelers now actively search for information using ICT in order to compose their vacation packages according to their specific emotionally driven preferences. Additionally when on-site, they search for freely available information about the site itself rather than renting a visitor guide that may be available, but considered to be expensive and sometimes outdated. However, like in many other cases, the blessing of the web comes with a curse; the curse of information overload. As such, recommender systems have been suggested as a practical tool for overcoming this information overload. However, because the tourism domain is extremely complex, those designing tourism-focused recommender systems face huge challenges.

This workshop brings together researchers and practitioners from different fields (e.g., tourism, recommender systems, user modeling, user interaction, mobile, ubiquitous and ambient technologies, artificial intelligence and web information systems) working in the tourism recommendation domain. The workshop aims to provide a forum for these people to discuss novel ideas for addressing the specific challenges for recommender systems in tourism with the goal to advance the current state-of-the-art in this field. Another goal of the workshop is to identify practical applications of these technologies within tourism settings from the point of view of individual users and user groups, service providers, as well as from additional stakeholders (e.g., destination management organizations). Finally, RecTour 2018 aims to continue the community building processes and discussions started at previous RecTour Workshops, i.e., at RecTour 2016 in Boston, MA, USA and at RecTour 2017 in Como, Italy.

Septermber 2018 — Julia Neidhardt, Wolfgang Wörndl, Tsvi Kuflik and Markus Zanker

Workshop Committees

Organizers

• **Julia Neidhardt**, TU Wien, Austria
• **Wolfgang Wörndl**, TU München, Germany
• **Tsvi Kuflik**, The University of Haifa, Israel
• **Markus Zanker**, Free University of Bozen/Bolzano, Italy

Program Committee

• **Catalin-Mihai Barbu**, University of Duisburg-Essen, Germany
• **Derek Bridge**, University College Cork, Ireland
• **Amra Delic**, TU Wien, Austria
• **Damianos Gavalas**, University of the Aegean, Greece
• **Ulrike Gretzel**, University of Southern California, USA
• **Wilfried Grossmann**, University of Vienna, Austria
• **Daniel Herzog**, TU München, Germany
• **Dietmar Jannach**, TU Dortmund, Germany
• **Jan Krasnodebski**, Expedia, Switzerland
• **Antonio Moreno**, Universitat Rovira i Virgili, Spain
• **Francesco Ricci**, University of Bozen/Bolzano, Italy
• **Hannes Werthner**, TU Wien, Austria
• **Zheng Xiang**, Virginia Tech, USA
• **David Zibriczky**, trivago, Germany

Workshop Program

9:00 - 10:30 Session 1: Workshop Opening and Keynote

• Workshop opening

• Keynote *Recommender Systems in Tourism: A Case for Interactive Approaches* by **Dietmar Jannach** (Alpen-Adria-Universität Klagenfurt, Austria)

10:30 - 11:00 Coffee Break

11:00 - 12:30 Session 2: Hotel Recommendation and Trip Planning

• **Marie Al-Ghossein, Talel Abdessalem and Anthony Barré**: *Cross-Domain Recommendation in the Hotel Sector.*

• **Catalin-Mihai Barbu and Jürgen Ziegler**: *Designing Interactive Visualizations of Personalized Review Data for a Hotel Recommender System.*

• **Daniel Herzog, Nikolaos Promponas-Kefalas and Wolfgang Wörndl**: *Integrating Public Displays into Tourist Trip Recommender Systems.*

• **Koji Kawamata and Kenta Oku.** *Roadscape-based Route Recommender System using Coarse-to-fine Route Search.*

12:30 - 14:00 Lunch Break

14:00 - 15:30 Session 3: Industry Perspective

• Keynote *User Experience and Data: Bridging the Gap* by **Themis Mavridis** (Booking.com, Netherlands)

• **Alejandro Mottini, Alix Lheritier, Rodrigo Acuna-Agost and Maria A. Zuluaga**: *Understanding Customer Choices to Improve Recommendations in the Air Travel Industry.*

15:30 - 16:00 Coffee Break

16:00 - 17:30 Session 4: Crowdsourcing and Evaluation

• **Öykü Kapcak, Simone Spagnoli, Vincent Robbemond, Soumitri Vadali, Shabnam Najafian and Nava Tintarev**: *TourExplain: A Crowdsourcing Pipeline for Generating Explanations for Groups of Tourists.*

• **Linus W. Dietz and Achim Weimert**: *Recommending Crowdsourced Trips on wOndary.*

• **Pablo Sánchez and Alejandro Bellogin:** *Challenges on Evaluating Venue Recommendation Approaches: Position Paper.*

• Workshop closing

Keynote *Recommender Systems in Tourism: A Case for Interactive Approaches*

by Dietmar Jannach, Alpen-Adria-Universität Klagenfurt, Austria

Abstract

There are various ways in which recommender systems can support their users in touristic contexts, from the selection of a destination, over pre-trip itinerary planning, to point-of-interest recommendation during the trip. In many of these application scenarios, building a recommender system solely on long-term interest profiles is not possible. Instead, interactive approaches are required in which users have the opportunity to interactively state and eventually revise their needs and preferences, and where the system is also able to explain its suggestions.

In this talk, we first review several application scenarios for recommender systems in tourism and summarize their specific requirements. We then focus on interactive recommendation approaches and specifically address the topics of explanations for recommender systems and mechanisms for user control.

About the speaker

Dietmar Jannach is a full professor of Information Systems at AAU Klagenfurt, Austria. Before joining AAU in 2017, he was a professor of Computer Science at TU Dortmund, Germany. In his research, he focuses on the application of intelligent system technology to practical problems and the development of methods for building knowledge-intensive software applications.

In the last years, Dietmar Jannach worked on various practical aspects of recommender systems. He is the main author of the first textbook on the topic published by Cambridge University Press in 2010 and was the co-founder of a tech startup that created an award-winning product for interactive advisory solutions.

Keynote *User Experience and Data: Bridging the Gap*

by Themis Mavridis, Booking.com, Netherlands

Abstract

Booking.com is the world's largest virtual two-sided marketplace with multi-dimensional, diverse and rich inventory. It aims to provide optimal shopping experience for our guests even with minimal user interaction. In this talk, I will discuss about the various unique and complex business cases, and the challenges that we face while we strive to bridge the gap between user experience and data. Finally, I will describe how we successfully utilize Machine Learning validated through rigorous Randomized Controlled Experiments in every step of the user journey.

About the speaker

Themis Mavridis is a Senior Data Scientist at Booking.com. In his work, he loves to apply machine learning, software engineering and statistics to customer-facing products. He leads the Machine Learning on Search at Booking.com. He is usually handling out-of-core / online algorithms for search, ranking and recommendations. Themis really enjoys "eating" Big Data with Vowpal Wabbit.

Furthermore, he used to do applied research on search engines, crawlers and topic modeling. His background is in Electrical & Computer Engineering. Themis studied at the Aristotle University of Thessaloniki (AUTH) in Greece and at the Vrije Universiteit (VU) Amsterdam in the Netherlands.

Table of Contents

Research Papers

Position Papers

Cross-Domain Recommendation in the Hotel Sector

Marie Al-Ghossein
LTCI, Télécom ParisTech
Paris, France
malghossein@enst.fr

Talel Abdessalem
LTCI, Télécom ParisTech
Paris, France
UMI CNRS IPAL NUS
talel.abdessalem@enst.fr

Anthony Barré
AccorHotels
Paris, France
anthony.barre@live.fr

ABSTRACT

Hotel recommendation suffers from a severe sparsity problem. Travelers only book hotels once or twice a year, and one booking dataset may not gather all the bookings done by one user. Cross-domain recommendation can be leveraged to face the sparsity problem by exploiting knowledge from a related domain where feedback can be easily collected. In this paper, we propose to leverage check-ins data from location based social networks to learn mobility patterns and use it for hotel recommendation, considering that the choice of destination is an important factor for hotel selection. We present our developed solution, where we map items and users from both domains based on a number of observations, learn preferences for regions and for hotels, and combine the results to perform the final recommendation. Experiments on a real booking dataset using a dataset of geolocated posts show the interest of using data from other domains to boost hotel recommendation.

1 INTRODUCTION

Recommendations in the travel and tourism domains have become essential with the exponential growth of available data on the Web which have turned trip planning into a tiring and time-consuming task [7]. In particular, hotel recommender systems (RS) help users in choosing an appropriate option for accommodation [2]. The high-stakes nature of selecting accommodations also leads to the necessity of guiding users when making a decision.

While RS have been deployed in many domains, hotel recommendation must take into account the constraints considered by users when choosing a hotel, which are not present in other domains. In addition, hotel recommendation suffers greatly from sparsity since traveling is not a frequent activity [4]. Users only travel a few times each year and the feedback collected is sometimes not enough to learn user preferences. Sparsity constitutes therefore a major limitation for collaborative filtering approaches.

One way to address the sparsity problem is to leverage knowledge from other related domains where it is easier to get information regarding the behavior of users. Cross-domain RS [8] take advantage of the abundance of heterogeneous data providing multiple views of users' preferences. They aim to improve recommendations in a target domain by exploiting preferences uncovered in source domains. When applied in the tourism domain, cross-domain RS can suggest, for example, hotels based on flight bookings or events to attend based on hotel bookings [3].

When organizing a trip, travelers usually select the destination to visit before choosing the hotel where they will stay and the choice of accommodation highly depends on its location. Choosing a destination to visit is in turn related to several factors. First, the majority of trips are meant to explore destinations which are close to the place of residence of travelers. Then, users tend to follow the actual trends running locally which are also likely to change with time. In addition, the timing of the trip has an impact on the chosen destination. Some destinations are more popular during summer than winter, and leisure trips are more frequent during vacation periods.

Since hotel bookings are collected by organizations managing a subset of hotels and accommodations, hotel booking datasets do not cover all trips done and destinations visited by users. On the other hand, recent years have witnessed the emergence of Location Based Social Networks (LBSN), e.g., Flickr and Foursquare, where the mobility of users is captured through their check-ins. When exploring points-of-interest, users share their experiences on LBSN, making them a rich data source to analyze travel experiences.

In this paper, we address the problem of hotel recommendation suffering from sparsity by leveraging check-ins data from LBSN. We learn mobility patterns from the check-ins which are easily shared on LBSN and use them in combination with hotel preferences in order to boost hotel recommendation. We first map check-ins and hotels to a common space of geographical regions based on the density of hotels spread worldwide. We learn preferences for geographical regions based on check-ins data, link users from both domains, and combine the preferences in order to generate recommendations. Experiments on a dataset of hotel bookings extracted from the hotel industry show the interest of using LBSN data.

The rest of the paper is organized as follows. In Section 2, we discuss related work on hotel recommendation and cross-domain recommendation. In Section 3, we present our approach for hotel recommendation leveraging mobility data from LBSN. Experiments and results are discussed in Section 4. Finally, Section 5 concludes the paper.

2 RELATED WORK

Hotel recommendation. Several data sources have been exploited in previous work to address the problem of hotel recommendation. Saga et al. [19] rely on *implicit feedback* in the form of booking transactions to build a hotel-user graph which is used as a preference transition network. Other proposed approaches consider *explicit feedback* in the form of textual reviews. Along this line, Levi et al. [12] use reviews written by users with similar background. Similarity is measured based on a set of criteria including the nationality, the travel intention, and preferences for hotel traits. Zhang et al. [24] use textual reviews to model users and hotels in latent topic spaces generating hotel and user similarity matrices. Nilashi et al. [15] leverage ratings on several aspects of hotels including the location, the cleanliness, and the value, among others. Hotel recommendation can also benefit from *contextual dimensions*. When addressing the problem of lodging recommendation, Sanchez-Vazquez et al. [20] consider several dimensions like the price sensitivity, the perceived value, and the risk involved in the selection, among others.

Even though destination is an important parameter for hotel selection, the problem of hotel recommendation is different than the one of point-of-interest recommendation [23]. Hotel visits occur while on trips that are separated by a return to the user's place of residence, while points-of-interest are frequently visited sequentially. To the best of our knowledge, this is the first work exploiting cross-domain information for the benefit of hotel recommendation.

Cross-domain recommendation. Cross-domain RS [8] aim to improve recommendation in a target domain by leveraging user preferences from a source domain. The main advantages of using cross-domain recommendation include diversifying recommendations, addressing the cold-start problem, and alleviating the sparsity problem. It is therefore possible to suggest songs to listen to based on users' preferences for movies, for example.

Passing from one domain to the other requires considering the overlap between users and items or the similarities between item features and user behavior, in the different domains. Cremonesi et al. [6] defines four scenarios for cross-domain recommendation derived from the overlapping possibilities of users and items: There could be no overlap between users and items from both domains, overlap between users, overlap between items, or overlap between both users and items.

Several techniques have been developed to perform cross-domain recommendation. Cantador et al. [5] presents a categorization of these approaches and distinguishes between two classes. The first one relies on aggregating knowledge collected from the various domains in order to perform recommendation. One way to do this is by merging user preferences in the form of ratings for example [14] or combining the recommendations from the various domains [9]. The second class of techniques manages to transfer knowledge from one domain to the other. This is done through sharing latent features [16] or transferring rating patterns [13].

In this work, we use basic approaches for cross-domain recommendation to leverage data from LBSN and we show the interest of using them for hotel recommendation.

3 OUR PROPOSED APPROACH

Motivation. In order to cope with the sparsity problem faced in hotel recommendation, we propose to learn mobility patterns from check-ins shared on LBSN and combine it with hotel preferences in order to generate recommendations. In the source domain $\mathcal{S}$, we have active users on LBSN, $\mathcal{U}_\mathcal{S}$, who share their check-in activity. The items $\mathcal{I}_\mathcal{S}$ are the geolocated points visited. The target domain $\mathcal{T}$ is the hotel domain where the users $\mathcal{U}_\mathcal{T}$ are the one booking hotels and the items to recommend, $\mathcal{I}_\mathcal{T}$, are the hotels. In the problem we are considering, there is no overlap between users from both domains as we are not able to link users posting on LBSN and users booking hotels. However, a mapping can be done between check-ins $\mathcal{I}_\mathcal{S}$ and hotels $\mathcal{I}_\mathcal{T}$ based on the corresponding location, and similarities between users from both domains, $\mathcal{U}_\mathcal{S}$ and $\mathcal{U}_\mathcal{T}$, can be computed based on the visited locations.

Our work is motivated by a number of ideas. First, our approach is inspired from the real decision-making process of users when choosing a hotel: They first select a destination to visit, and then a hotel where they can stay. The source domain contains users' paths through their check-in activity. We try to use the knowledge from the source domain to learn accessible destinations for users based on their history. Accessibility usually relies on distance, cost, value, and other hidden variables.

We are therefore interested in the mobility patterns at a high scale. In our problem, preferences for regions are more relevant than preferences for specific points-of-interest. Based on data from LBSN, we can get the set of cities visited by one user, for example, and use this information for recommendation. Once the destination is selected, the hotel choice is more likely to depend on its features. On the other hand, hotels are not equally spread worldwide. When considering regions where we have a high density of hotels, it would be relevant to learn preferences for different subregions. Since all the neighborhoods in a specific city do not have the same characteristics, travelers may prefer one over the other.

We consider therefore a decomposition of the world map in several regions, where the region size depends on the corresponding hotel density. Items from both domains, $\mathcal{I}_\mathcal{S}$ and $\mathcal{I}_\mathcal{T}$, are mapped to these regions. Using the behavior of users on LBSN, we learn the preferences of users to the defined regions.

To benefit from these insights, and since there is no overlap between users from both domains, we associate users from the target domain, $\mathcal{U}_\mathcal{T}$, with the ones that are the most similar from the source domain, $\mathcal{U}_\mathcal{S}$, with respect to the visited regions. Preferences for geographical regions and hotels are finally combined to generate hotel recommendations.

In the following, we detail each part of our approach. In this Section, a recommendation method designates any latent factor model [11] that can be used for uncovering latent factors representing users and items. A score is then computed for each hotel, and the items that get the highest score are proposed for recommendation.

Mapping items from both domains. As mentioned before, we decompose the world map into several regions denoted as $\mathcal{I}_{\mathcal{R}}$ and map check-ins and hotels to these regions based on their location. The world map decomposition depends on the density of hotels in each area: Regions with high density of hotels should be further decomposed into subregions.

We rely on a hierarchical division of the space into rectangular spaces used in [1] and inspired by the clustering approach STING [22]. The first level of the hierarchy covers the whole region considered and corresponds to the whole map which constitutes one cell. Each cell at a level l is partitioned into 4 cells at the next level $l + 1$, and the maximum number of levels is fixed.

We cluster hotels using a top-down approach based on the hierarchical structure of cells. The density of hotels in one cell is defined as the number of hotels located there divided by the area of the cell. For every level, starting with the first one, we compute the density of hotels in each cell. We then compare it to the density of the parent cell: If it is higher, we consider it as a cluster, otherwise, we move to the next level and repeat the process. This process is maintained until all the hotels are clustered or until we reach the maximum level.

Each cell containing a cluster of hotels is included in $\mathcal{I}_{\mathcal{R}}$. Each item from the sets $\mathcal{I}_{\mathcal{S}}$ and $\mathcal{I}_{\mathcal{T}}$ can be associated to a region from $\mathcal{I}_{\mathcal{R}}$. Using the feedback from the source domain (i.e., LBSN data) and $\mathcal{I}_{\mathcal{R}}$, we learn preferences for different regions.

Mapping users from both domains. Our aim is to use the preferences of users in $\mathcal{U}_{\mathcal{S}}$ to regions in $\mathcal{I}_{\mathcal{R}}$ to infer the preferences of users in $\mathcal{U}_{\mathcal{T}}$ to these regions. In order to do so, and for each user from $\mathcal{U}_{\mathcal{T}}$, we compute its neighbors (i.e., most similar users) contained in $\mathcal{U}_{\mathcal{S}}$ using a similarity measure. Z_u denotes the set of most similar users in $\mathcal{U}_{\mathcal{S}}$ for the user $u \in \mathcal{U}_{\mathcal{T}}$. The similarity measure handles user profiles from both domains defined as a binary vector which dimension is equal to the cardinality of $\mathcal{I}_{\mathcal{R}}$. If the user visited a check-in or a hotel located in a specific region, its value in the vector is 1, otherwise it is 0. We aggregate the region scores computed for each neighbor to get the scores for the target user.

Hotel recommendation. Performing hotel recommendation for a target user $u \in \mathcal{U}_{\mathcal{S}}$ requires computing hotels' scores, denoted by s_{ui}, for each hotel $i \in \mathcal{I}_{\mathcal{T}}$. Hotels are then ordered according to their scores and the k hotels having the highest scores are selected for recommendation.

The score computed is the combination of two scores: one from the source domain denoted by $s^{\mathcal{S}}_{ur}$, i.e., a score revealing the region preference for region $r \in \mathcal{I}_{\mathcal{R}}$, and the other from the target domain denoted by $s^{\mathcal{T}}_{ui}$, i.e., a score revealing the hotel preference for hotel $i \in \mathcal{I}_{\mathcal{T}}$.

In the source domain, we build a recommendation model modeling the preferences of users in $\mathcal{U}_{\mathcal{S}}$ to regions in $\mathcal{I}_{\mathcal{R}}$ and enabling the computation of scores of regions $r \in \mathcal{I}_{\mathcal{R}}$ for each user $z \in \mathcal{U}_{\mathcal{S}}$, i.e., $s^{\mathcal{S}}_{zr}$. In the target domain, we build a recommendation model modeling the preferences of users in $\mathcal{U}_{\mathcal{T}}$ to hotels in $\mathcal{I}_{\mathcal{T}}$ and enabling the computation of scores of hotels $i \in \mathcal{I}_{\mathcal{T}}$ for each user $u \in \mathcal{U}_{\mathcal{T}}$, i.e., $s^{\mathcal{T}}_{ui}$.

Final recommendations are performed for users from $\mathcal{U}_{\mathcal{T}}$. The score revealing the region preference for a user $u \in \mathcal{U}_{\mathcal{T}}$ is the aggregation of scores for the most similar users in $\mathcal{U}_{\mathcal{S}}$. The score revealing the hotel preference for a user in $\mathcal{U}_{\mathcal{T}}$, $s^{\mathcal{T}}_{ui}$, is directly computed using the built model. Both scores are combined and the final score for hotel i located in region r is given as follows, having a predefined weight parameter α:

$$s_{ui} = \alpha . s^{\mathcal{T}}_{ui} + (1 - \alpha) . \frac{\sum_{z \in Z_u} s^{\mathcal{S}}_{zr}}{|Z_u|} \tag{1}$$

In this work, we use a matrix factorization method, Bayesian Personalized Ranking [18], to learn preferences and compute scores since it performs well on our dataset of bookings. Any other recommendation method could be used within the same approach.

4 EXPERIMENTS

In this Section, we present the experiments we conducted to prove the interest of our approach.

Datasets. We used one dataset from each domain in order to test our approach. The hotel booking dataset is extracted from the hotel industry and contains bookings done by users during the last 3 years. It consists of 7.8M users, 4.5k hotels, and 34M bookings. Users come from all the world and hotels are spread in more than 90 countries.

We use YFCC [21], a real-world dataset published recently. It contains media objects which have been uploaded to Flickr between 2004 and 2014. A subset of the posts are annotated with geographic coordinates and can be used as check-ins. We consider users that have visited more than 5 regions from the one we define. The dataset we use contains around 24M check-ins done by 32k users.

Experimental setup. We split the booking dataset into a training and a test set. We sort the bookings of each user in a chronological order and select the first 80% of bookings as the training set and the rest as the test set. We also select 20% of the users who have only done one booking and add them to the test set in order to evaluate the performance on new users. We use the data from the training set to train our recommendation method and evaluate its performance on the test set.

Evaluation metrics. We consider that we recommend k hotels to each user and we note which of these hotels were actually visited by the user. We use $recall@k$ and $NDCG@k$ for measuring the performance. $recall@k$ is defined as follows:

$$recall@k = \frac{number\ of\ hotels\ the\ user\ visited\ among\ the\ top\ k}{total\ number\ of\ hotels\ the\ user\ visited}. \tag{2}$$

The Normalized Discounted Cumulated Gain (NDCG) measures the ranking quality and $NDCG@k$ is the normalized $DCG@k$ which is computed as follows:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{y_i} - 1}{\log_2(i+1)}, \qquad (3)$$

where y_i is a binary variable for the i-th hotel of the recommendation list, that is equal to 1 if the corresponding hotel is visited by the user and 0 otherwise. The $recall@k$ and $NDCG@k$ for the entire system are the average $recall@k$ and $NDCG@k$ over all evaluated users respectively.

Parameters. We performed a grid search over the parameter space of the methods in order to find the parameters that give the best performance. We report the performance corresponding to the parameters that lead to the best results.

Compared methods. We include in our comparison traditional recommendation methods that are listed in the following:

- **MostPop** recommends the most popular hotels to the users.
- **CB** is a content-based method where hotels and users are represented in the space of hotels' features using vector space models and tf-idf weighting [17]. Hotel features cover the location, the brand, the segment category, and offered services such as Wi-Fi connection, parking, meeting facilities, and children playground.
- **Knnu** is a user-centered neighborhood-based method where we use the Jaccard similarity measure and set the number of neighbors to 2000.
- **MF** is a matrix factorization technique handling implicit feedback [10]. We set the number of latent factors $K = 100$, the regularization parameters to 0.001 and $a = 1.0, b = 0.01$.
- **BPR** [18] is a matrix factorization technique that relies on pairwise preferences to learn the latent model. We set the number of factors $K = 100$ and the regularization parameters to 0.0025.
- **CD** is the method we propose in this paper, leveraging data from LBSN.

Results. Figure 1 shows the performance of the methods we consider. The results are represented for each category of users, defined by the number of bookings present in the training set. By definition, the metrics we use decrease when the number of bookings increases.

MostPop is the only method able to recommend hotels to inactive users (i.e., users with zero bookings in the training set). The inferiority of *CB* shows that users do not attribute a great importance to all the hotels' features considered. Further investigations showed that the location of the hotel is one of the few factors that greatly affect the decision. *Knnu* performs well for users with few bookings while *BPR* outperforms the other methods when the number of bookings increases significantly.

The results obtained for *CD* show the interest of using data from LBSN to alleviate the sparsity problem. *CD* outperforms all the other methods when the number of bookings is less than or equal to 10 bookings. The interest of using cross-domain information decreases when the number of bookings increases: *BPR* outperforms *CD* when the number of bookings is greater than 30.

One explanation may be due to the fact that the behavior of users actively sharing content on LBSN is not fully representative of the behavior of all travelers. In particular, people having done more than 30 bookings are more likely to be businesspeople which behavior is not necessarily similar to users on LBSN. In addition, once enough feedback about hotels is collected, it may be sufficient to learn hotel preferences and generate good recommendations. The interest of using *CD* is highlighted in the cold-start setting where hotel bookings alone are not enough to infer preferences. We note that the majority of users in the booking dataset have done less than 10 bookings and therefore, *CD* improves the global performance.

Discussion. This is the first work proposing to apply cross-domain recommendation in the hotel sector using, in particular, abundantly available data from LBSN. While it can be a promising approach especially in a sparse data environment, it opens several interesting challenges.

First, not all users have their mobility behavior represented by active users posting on LBSN. These users will not directly benefit from the proposed approach. A finer analysis of users posting on LBSN, i.e., users in the source domain, and users booking hotels, i.e., users in the target domain, may help identifying relevant user segments which behavior can be similar in both domains in terms of mobility and probably underrepresented in one or in both domains. As a first basic approach, we tried addressing this issue by defining segments based on the number of bookings made and comparing recommendation performances in each one, considering that the number of bookings made may reveal a certain aspect of the user category. Other alternatives including more advanced techniques may be applied. One possibility to benefit from cross-domain recommendation may be then to learn local models by user category.

Second, further advances in this direction should consider evaluating the recommendation diversity in terms of proposed locations. It is important to generate diverse recommendations and avoid suggesting hotels located in one same area. This may occur when one region gathering several hotels is promoted for a particular user.

Transferring knowledge from LBSN to the hotel sector may go beyond the mobility aspect by also considering temporality, i.e., periods during which one region is visited by specific users, and context of visits for example by analyzing metadata associated to the posts. In addition, while we used a clustering component to map both domains, other approaches for integrating knowledge may be exploited. For example, we may be considering to rely on a multi-task approach and to train models in both domains simultaneously.

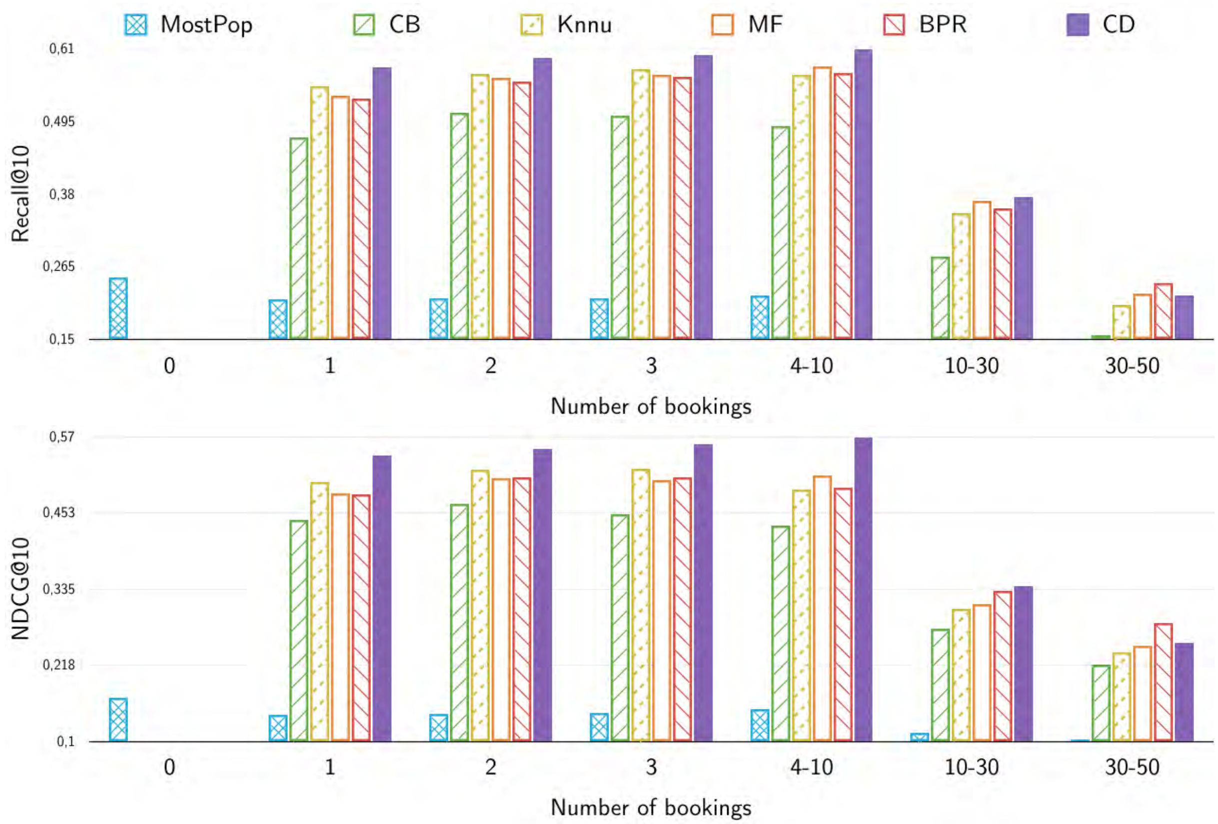


Figure 1: Recall@10 and NDCG@10 on the booking dataset. The results are represented based on the number of bookings in the training set.

5 CONCLUSION

In this paper, we propose to use data from LBSN to boost hotel recommendation. Hotel selection largely depends on the visited destination and some destinations are more accessible to users than others. Using the check-in activity from LBSN, we learn preferences for regions and use these preferences for hotel recommendation in order to address the sparsity problem. Mapping of items from both domains is done through a space of regions which definition is based on the density of hotels. Mapping of users from both domains is done by computing the similarity between users based on the visited locations. Hotel recommendation accounts for region preferences and hotel preferences. Experiments show the interest of using cross-domain information for users with few observations, i.e., in the cold-start setting.

Temporality plays an important role in the decision-making process: One destination is not considered by the same user in all periods of the year. Future work will involve adding the time dimension and taking into account in which period of the year the check-in was made in order to distinguish between users visiting the same destinations at different periods or seasons.

REFERENCES

[1] Marie Al-Ghossein and Talel Abdessalem. 2016. SoMap: Dynamic clustering and ranking of geotagged posts. In *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 151–154.

[2] Marie Al-Ghossein, Talel Abdessalem, and Anthony Barré. 2018. Exploiting Contextual and External Data for Hotel Recommendation. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP '18)*. 323–328.

[3] Marie Al-Ghossein, Talel Abdessalem, and Anthony Barré. 2018. Open data in the hotel industry: leveraging forthcoming events for hotel recommendation. *Information Technology & Tourism* (aug 2018). https://doi.org/10.1007/s40558-018-0119-6

[4] Lucas Bernardi, Jaap Kamps, Julia Kiseleva, and Melanie J. I. Müller. 2015. The Continuous Cold-start Problem in e-Commerce Recommender Systems. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems, Vienna, Austria, September 16-20, 2015*. 30–33.

[5] Iván Cantador and Paolo Cremonesi. 2014. Tutorial on Cross-domain Recommender Systems. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, New York, NY, USA, 401–402. https://doi.org/10.1145/2645710.2645777

[6] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. 2011. Cross-domain recommender systems. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. Ieee, 496–503.

[7] Alexander Felfernig, Sergiu Gordea, Dietmar Jannach, Erich Teppan, and Markus Zanker. 2007. A short survey of recommendation technologies in travel and tourism. (2007), 1722.

[8] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. [n. d.]. Cross-domain recommender systems: A

survey of the state of the art. In *Spanish Conference on Information Retrieval. sn, 2012.*
[9] Sharon Givon and Victor Lavrenko. 2009. Predicting Social-tags for Cold Start Book Recommendations. In *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. ACM, New York, NY, USA, 333–336. https://doi.org/10.1145/1639714.1639781
[10] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 263–272.
[11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.
[12] Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. 2012. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 115–122.
[13] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction.. In *IJCAI*.
[14] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *European conference on information retrieval*. Springer, 656–661.
[15] Mehrbakhsh Nilashi, Othman bin Ibrahim, Norafida Ithnin, and Nor Haniza Sarmin. 2015. A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS. *Electronic Commerce Research and Applications* 14, 6 (2015), 542–562.
[16] Weike Pan, Nathan N Liu, Evan W Xiang, and Qiang Yang. [n. d.]. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 2011*.
[17] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.
[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.
[19] Ryosuke Saga, Yoshihiro Hayashi, and Hiroshi Tsuji. 2008. Hotel recommender system based on user's preference transition. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. IEEE, 2437–2442.
[20] Raul Sanchez-Vazquez, Jordan Silva, and Rodrygo L.T. Santos. 2017. Exploiting Socio-Economic Models for Lodging Recommendation in the Sharing Economy. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. 260–268.
[21] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Commun. ACM* 59, 2 (2016), 64–73.
[22] Wei Wang, Jiong Yang, and Richard Muntz. 1997. STING : A Statistical Information Grid Approach to Spatial Data Mining. In *VLDB*.
[23] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 325–334.
[24] Kai Zhang, Keqiang Wang, Xiaoling Wang, Cheqing Jin, and Aoying Zhou. 2015. Hotel recommendation based on user preference analysis. In *Data Engineering Workshops (ICDEW), 2015 31st IEEE International Conference on*. IEEE, 134–138.

Designing Interactive Visualizations of Personalized Review Data for a Hotel Recommender System

Catalin-Mihai Barbu
University of Duisburg-Essen
Duisburg, Germany
catalin.barbu@uni-due.de

Jürgen Ziegler
University of Duisburg-Essen
Duisburg, Germany
juergen.ziegler@uni-due.de

ABSTRACT

Online reviews extracted from social media are being used increasingly in recommender systems, typically to enhance prediction accuracy. A somewhat less studied avenue of research aims to investigate the underlying relationships that arise between users, items, and the topics mentioned in reviews. Identifying these—often implicit—relationships could be beneficial for at least a couple of reasons. First, they would allow recommender systems to personalize reviews based on a combination of both topic and user similarity. Second, they can facilitate the development of novel interactive visualizations that complement and help explain recommendations even further. In this paper, we report on our ongoing work to personalize user reviews and visualize them in an interactive manner, using hotel recommending as an example domain. We also discuss several possible interactive mechanisms and consider their potential benefits towards increasing users' satisfaction.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Personalization*; • **Human-centered computing** → *User centered design*;

KEYWORDS

Recommender systems, Personalized reviews, Interactive visualization, Tourism, Multimode networks, Trustworthiness

1 INTRODUCTION AND MOTIVATION

As the research focus in recommender systems (RS) shifts gradually from prediction accuracy towards more user-centric methods, topics such as personalizing the user experience and increasing users' trust become more salient [11]. Transparency [18] and control [6] are frequently mentioned in the literature as important factors for achieving these goals. In this context, various approaches have been developed to support users in their exploration of recommendations. Collectively, these approaches are usually referred to as *interactive recommending* [7].

When many attributes need to be considered before making a choice, as is often the case in hotel RS, comparing ranked lists of recommendations often becomes cumbersome [3]. At the same time, alternative visualization techniques need to strike a fine balance with respect to the amount of information that can be presented while maintaining ease of understanding. Because of this inherent difficulty, ranked lists are frequently, despite their shortcomings, the preferred way to display recommendations. A promising middle-ground approach is to visualize specific aspects of a recommendation (e.g., user-generated content) while still retaining the traditional presentation style for the item lists. Prior research has established that online reviews can be a rich source of contextual information [2, 4, 26]. When presented alongside factual product attributes and standardized ratings, reviews can provide additional background evidence to support users in their decision-making process. Consequently, reviews are being used—with increasing effectiveness—as a further means of explaining recommendations [4, 20]. At the same time, large amounts of user-generated content also create an opportunity for personalization.

In this paper, we describe our ongoing approaches to personalize user reviews for a hotel RS and to visualize them in an interactive manner. The contribution of our work is threefold, namely to: 1) propose a model for identifying a suitable set of reviews to show a specific user, taking advantage of implicit relationships mined from those reviews; 2) develop methods to visualize said reviews to support users' decision-making; and 3) explore interactive mechanisms that allow users to maintain control over the visualization. Our approach builds upon the *co-staying* concept introduced in [1], wherein implicit multimode (user-topic-item) relationships extracted from user-generated content may be useful for increasing the trustworthiness of hotel recommendations.

In the following section, we report on the state of the art in review personalization and in information visualization techniques for RS. Afterwards, we present our conceptual model for personalizing reviews, using hotel recommendations as an example domain. Subsequently, we propose an approach for visualizing the data based on Sankey diagrams [24]. We also describe several mechanisms for interacting with the visualizations. Finally, we conclude by reflecting on our approach and enumerating promising directions for future research.

2 RELATED WORK

Although the importance of online reviews for explaining recommendations has been recognized in prior work (see, e.g., [20] for an overview), the topic of personalizing the presentation of reviews in RS has received relatively little attention from researchers. Moghaddam et al. [14] provides empirical evidence to support the fact that the perceived quality and helpfulness of online reviews differs across users. Their evaluation, which was performed on a real-life dataset of reviews, compared two latent factor models for predicting the personalized review quality. Similarly, Tu et al. [21] aim to personalize the set of reviews shown to users by decreasing redundancy and maximizing the coverage of topics of interest. Once a suitable set of reviews has been identified, the next challenge is how to present them.

Information visualization for RS is an active and promising field of research [10]. Several approaches have been proposed for visualizing recommendations in an interactive manner. We believe some of these approaches could also be adapted for visualizing specific aspects of a recommendation. In *SetFusion* [16], a hybrid RS for conference talks, the authors enhance the typical ranked list paradigm with interactive Venn diagrams. The charts afford users a new perspective on examining and filtering recommendations. The implementation is a successor of *TalkExplorer* [23], in which the relevant information was represented using cluster maps. Yazdi et al. [25] propose a bubble graph representation for suggesting collaboration opportunities. They show that the visualization helps users form a mental model of the recommendation space and the connections between scholars, institutions, and research topics. A similar visualization metaphor is also used in [15] to recommend contacts in social networks. Richthammer and Pernul [17] employ treemapping to facilitate users' exploration of movie recommendations. They show that the structured presentation makes it easier for users to obtain an overview of the search space and possible alternatives. In contrast, Kunkel et al. [12] render the movie domain space on a 3D map that can be reshaped by users to "uncover" similar recommendations. Finally, Tietz et al. [19] proposes a method for visualizing multimedia content based on linked data. Displaying the semantic relationships graphically supports exploration and the discovery of new content.

Despite the multitude of techniques, most of them are inherently limited in the number of elements that can be realistically depicted on a screen. Thus, identifying and grouping items into clusters becomes a key requirement for reducing clutter and helping users cope with the amount of information. Several approaches have been proposed in the field of social network analysis that can be applied to multimode networks (see, for instance, [8], [9], and [13]).

3 PERSONALIZE A SET OF HOTEL REVIEWS

Whether a hotel review is considered helpful by a user may depend on several aspects, among them individual preferences (e.g., "I prefer to sleep on a soft mattress; what have previous guests written concerning bed quality?"), the specifics or requirements of the travel scenario (e.g., "I am traveling for work, so I am mostly interested in the opinions of other business travelers."), and various sociodemographic factors (e.g., "What do people who, like me, usually book 3-star hotels think about these accommodations?"). The goal of personalization is to show users the most relevant reviews, based on their recorded preferences [21]. Our hypothesis is that both the content of the review and metadata about the person who wrote it can be leveraged to calculate a relevance score. This would allow a RS to prioritize hotel reviews that: 1) mention the topics in which the user is interested; and, at the same time, 2) are written by people who have the most in common with the user.

Various techniques have been proposed for extracting features and user attitudes from online reviews [2, 4, 26]. Most commonly, the output is a list of concepts, or topics, that appear often in reviews (for example, "soft bed" or "quiet room"). User attitudes about a certain topic can be either positive, negative, or neutral [26]. In [1], we described how the connections between users, hotels, and topics form an *implicit* social network—meaning that users do not communicate directly with each other. Instead, relationships are formed based on the hotels that they have visited in the past and the topics that they mentioned in their reviews.

Figure 1: Eliciting user preferences. Users can drag and drop relevant topics from the categories on the left-hand side to the "Preferences" area on the right-hand side. Sliders can be used to adjust the importance of each attribute.

For the sake of simplicity, and as an initial step towards testing our hypothesis, we decided to elicit user preferences as part of the recommendation process. Concretely, in our application—which is based on the one described in [5]—users shall be asked to select (and assign weights to) hotel characteristics that are most important to them (Figure 1). This interaction bears similarities to how a person typically interacts with online booking portals: After choosing a destination and travel date, users are normally presented with a list of filters that they may use to refine the list of recommendations even further. Clicking on a filter labeled "beach", for instance, will prioritize hotels located near the seafront. Such an action can be regarded as preference elicitation. In our prototype, we feed this information into the RS not only to find recommendations, but also to personalize the reviews.

Once they have been elicited, user preferences can be matched against pre-extracted topics (see [5]) to select the most suitable reviews. For each review belonging to one of the recommended hotels, a partial relevance score, R_c, can be computed based on the number of topics that match the user preferences. A second, and arguably more interesting step, is to additionally consider user similarity when calculating a review's final relevance score. We identified four user factors that we consider relevant for this task. A reviewer's *rating behavior* denotes the extent to which her hotel scores match those of other users who share similar preferences. This is, in essence, the basis for collaborative filtering [11]: For a given set of hotels, we expect like-minded guests to give more homogeneous ratings. The *travel profile* represents a combination of aspects that characterize the reviewer's typical hotel booking. These may include the purpose of travel (i.e. business or leisure), room type, number of nights, time of year etc. Another factor is the degree to which a reviewer's own *set of preferences* is well-defined. For example, reviews contributed by someone who often gives feedback on the quality of the bed are probably more relevant to a user who cares about this aspect of a hotel room. Finally, we check whether the reviewer has stayed in *similar hotels*. For this, we consider both objective information, such as a hotel's star rating, and prevalent topics extracted from user-generated content. Prior work suggests

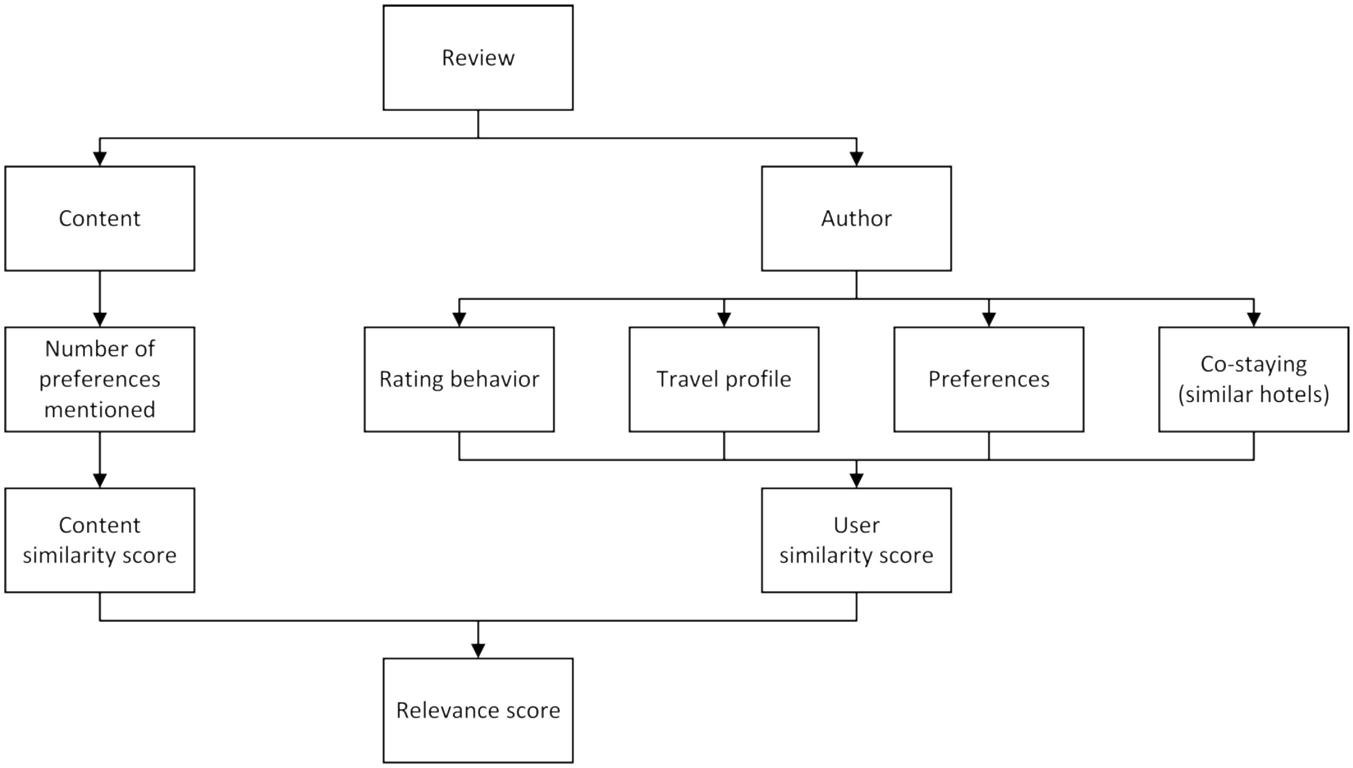


Figure 2: Proposed model for calculating the relevance score of a review, taking into account both its content and its author.

that people who book similar hotels may also have comparable expectations [2, 21]. By combining these factors, the second partial relevance score, R_u, can be calculated. The review's final relevance score can be written as $R_r = R_c \cdot wc + R_u \cdot (1 - wc)$, where the weighting factor wc will be found empirically. An overview of the proposed model is shown in Figure 2. With the exception of travel profile, all factors can be extracted from information contained in the co-staying network. The remaining factor can be obtained from the reviews' metadata (i.e. the review date and automatically-generated tags about the hotel booking, such as duration of stay, type of room, and number of guests). The dataset used for generating the co-staying network is the one described in [1].

As a further refinement, we will also explore the possibility of using reviews written for hotels that are part of the same chain as the recommended hotels. Our premise is that hotel chains typically strive to achieve a consistent user experience across their sites [1]. This means that, per our *co-staying* concept, two reviewers can be considered similar even if they previously booked rooms in different locations of the same hotel franchise. We aim to evaluate our approach by comparing it against latent factor models, such as the one suggested in [14]. We believe the additional relationships captured by the multimode network will yield improved results when compared to other review personalization approaches.

4 VISUALIZE AGGREGATED REVIEW DATA

Based on our review of the literature (see section 2), we believe there is significant potential in combining traditional RS with a means to explore information related to a specific hotel recommendation in a more visual manner. Concretely, we started developing graphical representations of relevant hotel topics (and their authors) based on: 1) how often they appear in the user-generated content; and 2) their valence (i.e. positive or negative mentions). To avoid information overload, we purposefully restrict the visualization to only a personalized set of hotel reviews, as identified in the previous section. Our aim is to find out whether such a visualization has a significant effect in terms of helping users understand better why a hotel was recommended. Thus, we consider the visualization as an additional form of explanation. Constraining the visual representation to relatively small amounts of data (i.e. from a personalized subset of reviews) also alleviates the main shortcoming identified in the related work section. At the same time, we believe our approach remains in line with the typical use cases of hotel RS. Specifically, most people have a limited number of preferences (i.e. topics) in which they are interested in for a given trip.

We experimented with two graphing methods, namely: 1) Treemap, an area-based visualization [17]; and 2) Sankey, a type of flow diagram [24]. Both techniques have specific advantages and shortcomings. In general, Treemaps provide a good overview, but users might find it more difficult to focus on specific details. In contrast, Sankey diagrams tend to have a higher legibility. This is due to their flow structure, which generally follows a left-to-right (or, less frequently, top-to-bottom) orientation that might be easier for users to grasp. Because of this aspect, we will focus on Sankey visualizations in the remainder of this paper.

The layout of a Sankey diagram is flexible enough to accommodate multiple levels of nodes. As a result, it is well-suited for

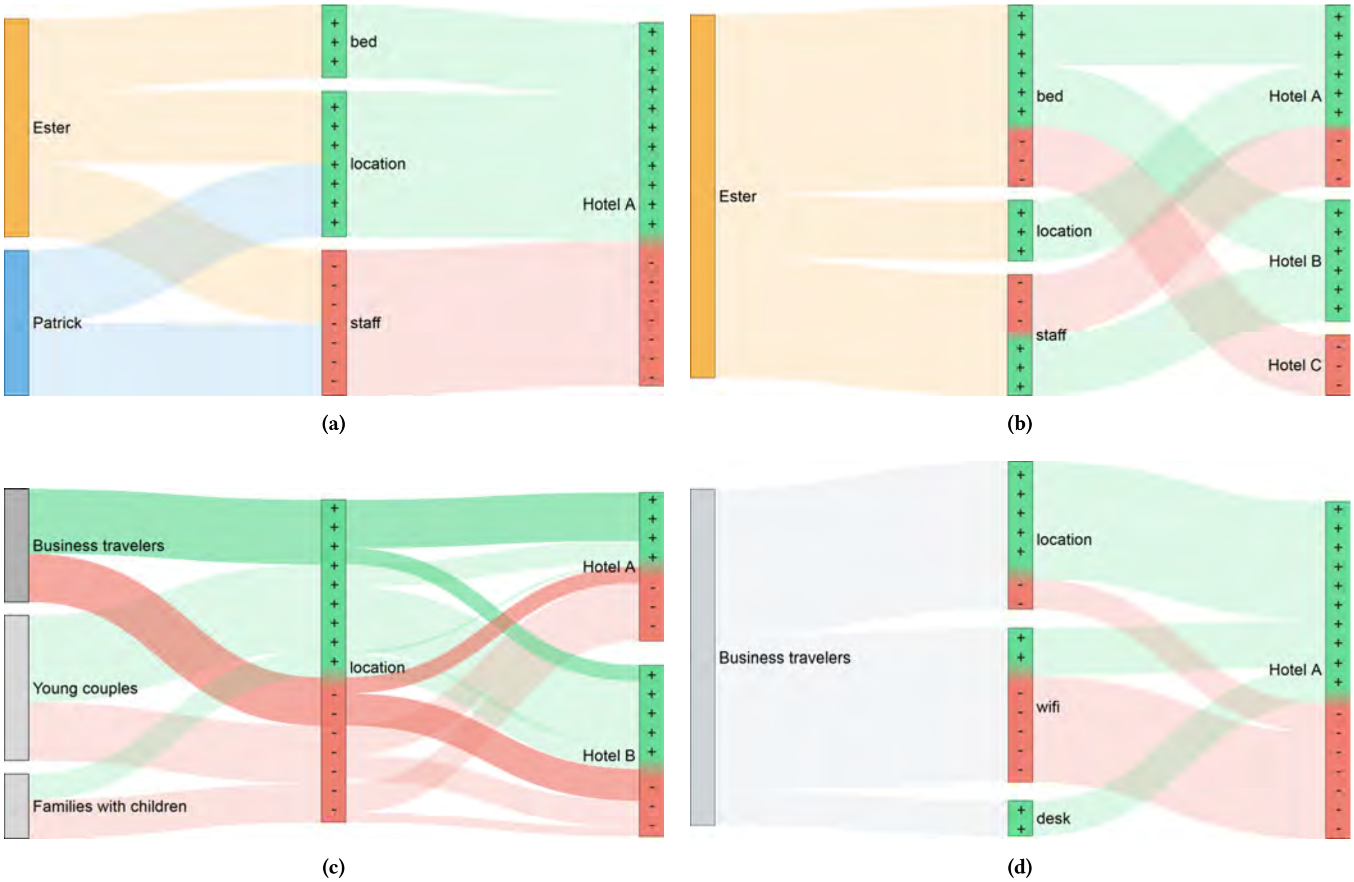
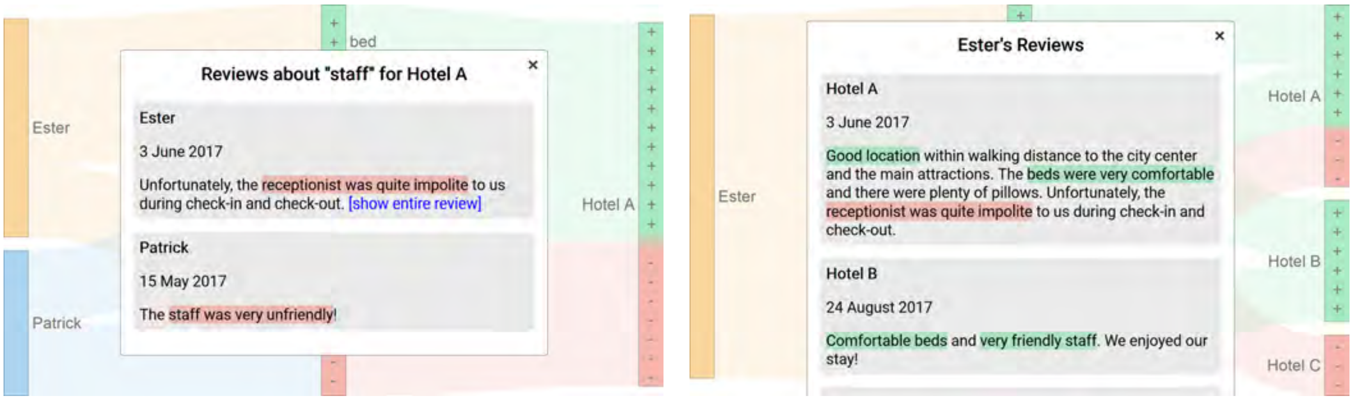


Figure 3: Example visualizations using Sankey diagrams. Different colors (green and red) and symbols ("+" and "-") are used to denote positive and negative mentions, respectively. Top left: Topics mentioned by two users in their reviews about Hotel A. Top right: Topics mentioned by a user in her hotel reviews. Bottom left: Opinions regarding the location of two hotels have been aggregated based on users' travel category. Links originating from the group "business travelers" are highlighted. Bottom right: Subset of topics mentioned by a group of users who reviewed Hotel A.

visualizing multidimensional data, such as the user-topic-hotel relationships that form the backbone of our co-staying network. Four typical visualizations are shown in Figure 3. Each follows a similar pattern, with the user (or user group) nodes placed on the left, topic nodes in the middle, and hotel nodes on the right. Edges between nodes correspond to topic mentions; the width of an edge is proportional to the number of times its corresponding topic appears in a user's reviews. User sentiment is represented using colors (i.e. red and green for negative and positive mentions, respectively) and symbols (i.e. "-" for negative and "+" for positive mentions). Furthermore, the coloring of topic and hotel nodes indicates the proportion of positive vs. negative references. These graphical elements are meant to help users perceive quickly the prevailing user sentiment on a given issue. Specific paths in the Sankey diagram can be highlighted to increase their salience, as shown in Figure 3c. As depicted in Figure 3c and Figure 3d, the visualization can also be used to compare two or more hotels.

Since many prospective users might not be familiar with Sankey diagrams, we formulate several interactive mechanisms to support them. First, and most importantly, users should be able to control the amount of information that is represented in the chart. One way to achieve this is by clustering nodes to reduce clutter and increase legibility. This is especially relevant in the case of user nodes, which will almost always be the most numerous of the three vertex types. A relatively straightforward possibility is to group users based on whether they are traveling for business or leisure (Figure 3c). A more interesting approach that we are investigating is how to cluster users based on their similarity scores, which are computed using the algorithm described in the previous section. Furthermore, topics can also be clustered, for example based on whether they refer to the hotel in general (e.g., "location"), a room feature (e.g., "shower"), or the quality of the service (e.g., "staff").

Users will also have the option to "zoom" in or out in order to fine tune the level of detail. Another way to control the visualization is by providing adequate filtering mechanisms. For example, the user may select only a subset of topics to visualize, or she might decide to view only topics with negative opinions. Even so, showing all three layers of the underlying multimode network at once might still prove too difficult for some users to comprehend. Therefore, one possible solution is to limit the visualization to only two types of



(a) (b)

Figure 4: Reviews are displayed (on demand) as a separate layer on top of the Sankey visualization. Topics are highlighted according to their valence. Left: Users' opinions about a particular topic related to Hotel A. For the top review in the list, only a relevant snippet is shown. Right: Partial view of the reviews written by a user.

vertices. In this case, suitable interface elements could be provided to facilitate interaction with the third dimension, e.g., by using filters.

Clicking on the nodes also affords interesting interaction opportunities. One example is to allow users to "refocus" the visualization around a specific node. In Figure 3a, clicking on one of the two users changes the diagram to show only the topics mentioned by that user (Figure 3b). Similarly, selecting a topic would display only the users who referred to that topic in their reviews. Finally, clicking on the hotel would have the effect of reverting to the default visualization. An interesting open question, which we plan to verify empirically, is whether to allow users to reorganize the diagram by dragging and dropping nodes. Such functionality may facilitate "ad-hoc" clustering. Moreover, the resulting arrangement could also be saved as a template, so that future visualizations are rendered, by default, in a similar fashion.

Initially, our Sankey diagram implementation does not display the actual content of the reviews. However, users can easily access this information on demand (cf. Figure 4). One relatively simple method to achieve this functionality is to render the appropriate reviews in an overlay window. The content and presentation style are determined by the node or edge with which the user interacted. In Figure 4a, interacting with the node "staff"—e.g., by double-clicking—displays users' feedback on that topic. (Note that the underlying Sankey diagram is identical to the one in Figure 3a.) Furthermore, the top review in the aforementioned example has been condensed to a relevant snippet; however, the user may toggle an embedded link to view the entire text. By the same token, interacting with either a hotel or with a user node depicts all hotel reviews, or the opinions contributed by a specific user, respectively. An example of the latter is shown in Figure 4b (see also Figure 3b for the initial visualization). Moreover, this type of interaction is implemented for edges as well. Alternatively, a user may only be interested in finding out quickly how many times a topic has been mentioned, without perusing the reviews. In this case, simply hovering over an edge will display this information in a summarized form, e.g., "'breakfast' ⟶ 5 mentions (mostly positive)".

5 DISCUSSION AND FUTURE WORK

As the amount of user-generated content continues to grow, it is becoming increasingly important to develop methods for filtering and personalizing the content used for explaining recommendations. We propose a model for identifying personalized sets of reviews in a hotel RS, which combines both content and user similarity to calculate a relevance score for each review. In particular, we believe that better user similarity measures can be developed by taking into account ternary relationships such as those in our *co-staying network* [1]. Specifically, we are investigating connections between travelers who: 1) booked the same hotel(s); 2) stayed in similar hotels (e.g., that are part of the same chain); 3) have a well-defined set of topics that they mention frequently in their reviews; and 4) exhibit a similar rating behavior. Furthermore, we suggest a method for displaying a subset of personalized reviews graphically using Sankey diagrams. Allowing users to explore the multimode relationships could be considered as an additional form of explaining recommendations [20]. As future work, we aim to evaluate empirically whether these approaches, combined, increase users' understanding of the reasons behind recommending a specific hotel. We expect that such an outcome would, in turn, have a positive effect on the transparency and perceived trustworthiness of hotel RS.

Although not specifically discussed in this paper, methods for visualizing user opinions could be of interest also to hotel managers. In combination with interactive mechanisms, such as the ones suggested in the previous section, these graphical representations could provide a clearer picture of the feedback that guests typically write. This could help monitor and focus on areas that require improvement, i.e. topics with numerous negative mentions. The usefulness of these methods in other domains, such as data analytics or visualization RS [22], should also be investigated further.

ACKNOWLEDGMENTS

This work is supported by the German Research Foundation (DFG) under grant No. GRK 2167, Research Training Group "User-Centred Social Media".

REFERENCES

[1] Catalin-Mihai Barbu and Jürgen Ziegler. 2017. Co-Staying: A social network for increasing the trustworthiness of hotel recommendations. In *RecTour '17*. CEUR-WS Vol. 1906, 35–39.

[2] Guanliang Chen and Li Chen. 2015. Augmenting service recommender systems by incorporating contextual opinions from user reviews. *User Modeling and User-Adapted Interaction* 25, 3 (2015), 295–329.

[3] Cecilia di Sciascio, Vedran Sabol, and Eduardo E Veas. 2015. uRank: Exploring document recommendations through an interactive user-driven approach.. In *IntRS '15*. CEUR-WS Vol. 1438, 29–36.

[4] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2018. Explaining recommendations by means of user reviews. In *ExSS '18*. CEUR-WS Vol. 2068.

[5] Jan Feuerbach, Benedikt Loepp, Catalin-Mihai Barbu, and Jürgen Ziegler. 2017. Enhancing an Interactive Recommendation System with Review-based Information Filtering. In *IntRS '17*. CEUR-WS Vol. 1884, 2–9.

[6] F Maxwell Harper, Funing Xu, Harmanpreet Kaur, Kyle Condiff, Shuo Chang, and Loren Terveen. 2015. Putting users in control of their recommendations. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 3–10.

[7] Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications* 56 (2016), 9–27.

[8] Dmitry I Ignatov, Alexander Semenov, Daria Komissarova, and Dmitry V Gnatyshak. 2017. Multimodal Clustering for Community Detection. In *Formal Concept Analysis of Social Networks*. Springer, 59–96.

[9] Isaac Jones, Lei Tang, and Huan Liu. 2015. Community discovery in multi-mode networks. In *User Community Discovery*. Springer, 55–74.

[10] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 441–504.

[11] Joseph A Konstan and John Riedl. 2012. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 101–123.

[12] Johannes Kunkel, Benedikt Loepp, and Jürgen Ziegler. 2017. A 3D item space visualization for presenting and manipulating user preferences in collaborative filtering. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 3–15.

[13] Bo Long, Zhongfei Mark Zhang, Xiaoyun Wu, and Philip S Yu. 2006. Spectral clustering for multi-type relational data. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 585–592.

[14] Samaneh Moghaddam, Mohsen Jamali, and Martin Ester. 2011. Review recommendation: personalized prediction of the quality of online reviews. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 2249–2252.

[15] Sayooran Nagulendra and Julita Vassileva. 2014. Understanding and controlling the filter bubble through interactive visualization: a user study. In *Proceedings of the 25th ACM International Conference on Hypertext and Social Media*. ACM, 107–115.

[16] Denis Parra, Peter Brusilovsky, and Christoph Trattner. 2014. See what you want to see: visual user-driven approach for hybrid recommendation. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*. ACM, 235–240.

[17] Christian Richthammer and Günther Pernul. 2016. Explorative analysis of recommendations through interactive visualization. In *International Conference on Electronic Commerce and Web Technologies*. Springer, 46–57.

[18] Rashmi Sinha and Kirsten Swearingen. 2002. The role of transparency in recommender systems. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*. ACM, 830–831.

[19] Tabea Tietz, Jörg Waitelonis, Joscha Jäger, and Harald Sack. 2014. Smart Media Navigator: Visualizing recommendations based on linked data. In *International Semantic Web Conference (Industry Track)*. CEUR-WS Vol. 1383.

[20] Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender Systems Handbook*. Springer, 353–382.

[21] Wenting Tu, David W Cheung, and Nikos Mamoulis. 2017. More focus on what you care about: Personalized top reviews set. *Neurocomputing* 254 (2017), 3–12.

[22] Manasi Vartak, Silu Huang, Tarique Siddiqui, Samuel Madden, and Aditya Parameswaran. 2017. Towards visualization recommendation systems. *ACM SIGMOD Record* 45, 4 (2017), 34–39.

[23] Katrien Verbert, Denis Parra, and Peter Brusilovsky. 2014. The effect of different set-based visualizations on user exploration of recommendations. In *IntRS '14*. CEUR-WS Vol. 1253, 37–44.

[24] Krist Wongsuphasawat and David Gotz. 2012. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2659–2668.

[25] Mohammad Amin Yazdi, André Calero Valdez, Leonhard Lichtschlag, Martina Ziefle, and Jan Borchers. 2016. Visualizing opportunities of collaboration in large research organizations. In *International Conference on HCI in Business, Government and Organizations*. Springer, 350–361.

[26] Yongfeng Zhang. 2015. Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*. ACM, 435–440.

Recommending Crowdsourced Trips on wOndary

Linus W. Dietz
Technical University of Munich
Garching, Germany
linus.dietz@tum.de

Achim Weimert
wOndary LTD
London, United Kingdom
achim@wondary.com

ABSTRACT

Making recommendations for tourist trips is a challenging task due to the intrinsic complexity of the domain. The characterization of itineraries is non-trivial, because there is a lack of open destination databases such as regions, islands, cities or attractions that would help to understand the characteristics of destinations within a trip. For this purpose, we present wOndary, which supports the planning and sharing of worldwide trips based on crowdsourcing. We sidestep item discovery and routing challenges of the tourist trip design problem by performing content-based recommendation by facilitating a novel structured representation of itineraries. We share our experiences in the establishment of the core model for our travel recommender system and discuss future developments.

KEYWORDS

recommender systems, user modeling, crowdsourcing, explaining recommendations, critiquing

1 INTRODUCTION

Independent travel planning is very complex. Today's connected world offers a myriad of choices of where to travel to, and there is unlimited information based on which one can make a decision. wOndary[1] has developed a platform for independent travelers to plan their trips. Initially, it started as a planning tool to create personal itineraries that can be shared privately with friends and co-travelers or that can be published as a public trip on the platform. In this paper, we describe how we transition the wOndary platform to a personalized recommender system for crowdsourced trips and describe the future potential of this work.

Our proposed solution involves the following contributions. We present a data model for structuring trips into blocks that are both useful for users and for segmenting trips. Furthermore, we present an attractions categorization that enables content-based recommendations via implicitly elicited preference. Utilizing this novel data model for structured itineraries, we provide recommendations for complete trips, and for parts of trips, i.e., blocks. The approach was driven by the following research questions (RQs):

RQ 1: What is a suitable recommendation model that masters the complexities of travel and enables future innovation regarding the user experience within wOndary?

RQ 2 How can crowdsourced trips be structured and characterized to enable content-based recommendations?

RQ 3: How can user preferences be elicited without requiring much effort by the user?

In the following section, we describe wOndary and the core of the novel content-based travel recommender. Then, in Section 3 we survey prior literature on personalized travel recommendation and further discuss avenues to improve the current basic system in Section 4. We conclude this paper in Section 5.

[1]https://wondary.com

2 TRAVEL RECOMMENDATION FOR INDEPENDENT TRAVELERS

wOndary is a platform that allows users to save, organize and share details about their trips. The platform helps with the structuring of personal itineraries, enables collaboration between group travelers, and encourages the publishing of personal itineraries so that others can reuse and customize these crowdsourced trips for their own purposes. wOndary currently focuses on young urbans (23–30 year olds) that strive for unique experiences during independently planned trips.

The user journey on wOndary reflects the *travel micro-moments* as defined by Google as "*dreaming*, *planning*, *booking*, and *experiencing*" [11]. When users dream of going away, they browse crowdsourced itineraries on wOndary or read travel-related stories. Online travel media, such as travel blogs can include wOndary's widget to refer users to unique itineraries that have been created by other travelers. In the planning phase the users save activities or copy itineraries to quickly create their own, customized trip. The users can search for specific locations and activities on and off the platform and collaborate with their co-travelers. By synchronizing the wOndary itinerary to the calendar app on their phone, the trip info becomes available when a user is offline to experience the foreign culture, but can be adapted at any time if there is Internet connectivity. Once the users return from their trip, they can privately share their itinerary with friends and colleagues or decide to publish it to all other users within the platform.

wOndary features a web application that is currently available in open beta on https://wondary.com. It is implemented as a single-page-application that runs on the Google Cloud Platform, and therefore, works in web browsers on all types of devices.

2.1 Finding Inspiration with the "Explore" Page

The users need a structured way to access the growing number of crowdsourced itineraries. To answer our first research question, wOndary provides a location-based *"Explore"* feature that allows querying a location, and filtering by geographical bounds and attributes, such as trip duration. Filtering is possible by adjusting the trip duration (72 hours, 1 week, or 2+weeks), the season, and the query area by adjusting the map excerpt. We chose this visual representation for the recommendations because a complex domain, like global travel, requires an intuitive user interface instead of a simple list. Therefore, matching items are displayed on a map and as a list, where their ranking depends on the distance to the queried location constituting the baseline for future improvements.

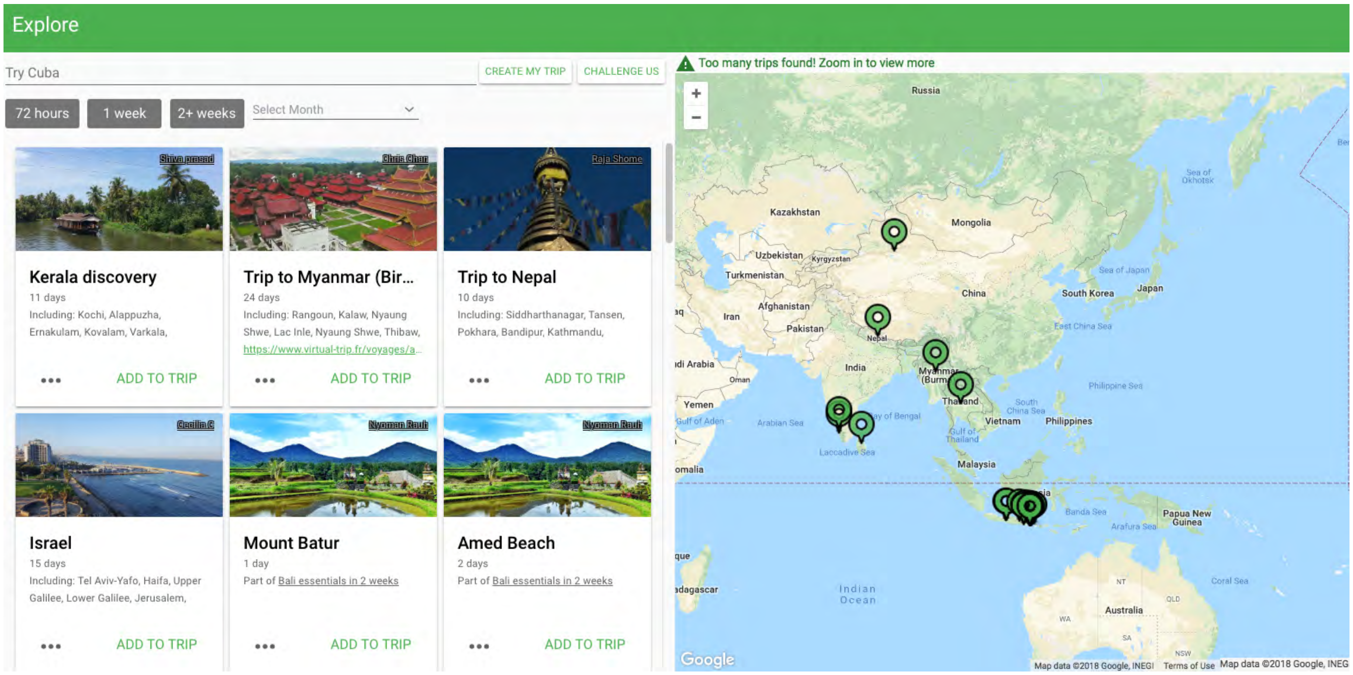


Figure 1: Explore Page, https://wondary.com/explore

The recommended items are both full itineraries as well as blocks, as defined below. As can be seen in Figure 1, the interface allows users to zoom into a geographical region. The users can also view high-level information about itineraries and focus on the ones they would like to see further details about, or they can copy them as a basis for their own customized trip.

With an increasing number of trips being published on the platform, it has become difficult for users to identify itineraries that fit their travel requirements. For example, itineraries are diverse in terms of included activities, and a user who loves sightseeing may not be interested in a trip that features primarily beaches or a multi-day hike through the mountains. Additionally, the users expect websites to support them in finding relevant content. Last, the number of trips for a popular region make it tedious for users to review all itineraries. Therefore, recommendations are playing an increasingly important role in wOndary's Explore feature because showing relevant content to the user improves their engagement and general satisfaction with the application.

2.2 A Data Model for Structured Itineraries

wOndary's data model for trips is based on insights from the domain. When travelers plan their trips, they often think of destinations, e.g., cities that they want to connectively visit. For example, a trip to Italy would start with several days in Rome, then, a day in Florence, visiting friends in Bologna over the weekend, and finish with three more days in Venice. To capture this, wOndary structures trips into blocks. A block acts as a descriptor of a partial trip that has a duration of one or more consecutive days and links to a location. Thus, trips are modeled as a sequence of one or more blocks. This structure was designed using user feedback and matches the way travelers approach planning. Additionally, it allows the normalization of trips spanning longer periods of time (several weeks or months) into portions that are transferable between trips of different travelers. wOndary heavily relies on blocks, not only when recommending items but also when presenting structured information about trips to users.

The next lower level of the data model is the day, consisting of three types of entries: transportation, lodging, and activities. Having a good overview of how to get from one place to another and where to stay overnight is essential for planning travel, whereas, instead, travelers define their trips based on the attractions they visit during the day. Currently, the users can input the attractions using venues from Google Places to ensure that they actually exist; typos are corrected, and duplicates are eliminated. Furthermore, the Google Places service provides further information, such as an image, opening hours, or ratings.

To perform content-based recommendations, it is necessary to classify items and the users into some meaningful categories. Therefore, our answer to RQ2 is the aforementioned data model using the five categories listed below, which are influenced by the target audience of the platform and the available attraction information. We compiled them based on an analysis of the platform's trips combined with our expert knowledge on individual travel.

- **Food** Mainly comprises restaurants and cafés, but also grocery stores and food markets.
- **Culture** Describes activities and places with cultural or historical attributes. For example, museums, galleries, churches and theaters fall under this category.
- **Nightlife** Categorizes places that are commonly related to nightlife such as bars, night markets, and jazz clubs.

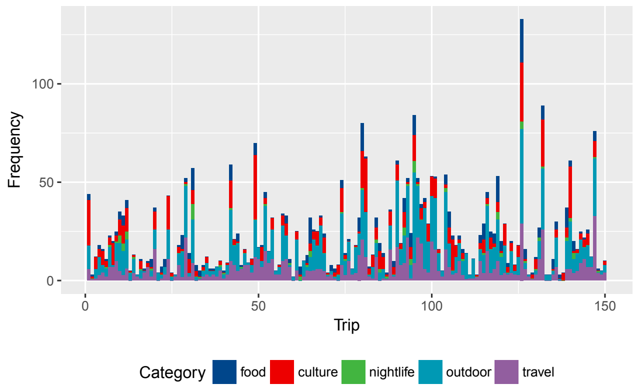


Figure 2: Frequencies of Categories per Trip

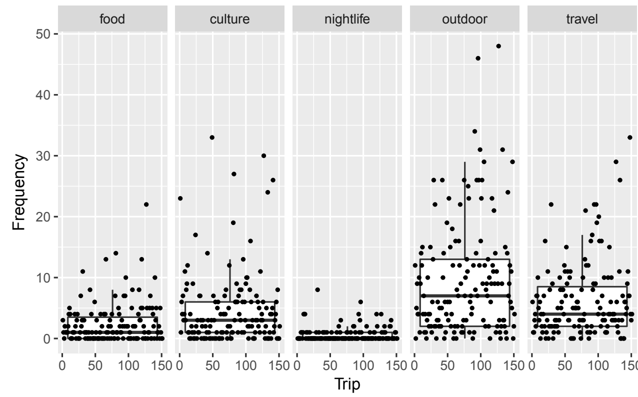


Figure 3: Classification of Trips per Category

- **Outdoor** Includes attributes associated with natural scenery or outdoor activities, such as parks, nature preserves, beaches, mountains and trails.
- **Transport & Travel** Consists of travel-related attractions, such as ferries, train stations and airports. This indicates that a relevant portion of the day is spent on transportation and that the transfer itself is an attraction.

2.3 Content-Based Travel Recommendations

To categorize the attractions, we query the Google Places types[2] and directly map them into our five categories. However, the returned place types are not primarily meant for travelers. For example, the type query for the Colosseum of Rome, Italy returns:

```
"types": ["point_of_interest", "establishment"]
```

While these types are not totally off mark, the information is insufficient to categorize this monument into one of our categories. Therefore, we augment the types from Google with an additional lookup of the attraction via the Foursquare API to allow one attraction to be a member of several categories. Foursquare has a rich hierarchical region categorization[3] with 923 categories that are organized in a tree to model specialized subcategories. To locate a Google Place on Foursquare, we performed a query by name using the exact location. By doing a bulk comparison, we found that most attractions also exist in Foursquare, except for political entities, such as city names. Conveniently, due to the bounded local search, the first result for Foursquare was the correct result for the corresponding Google Place. Recalling our example, we found that Colosseum was categorized as a *"Historic Site"*, which is within the *"Arts & Entertainment"* category of Foursquare. Using static mapping of all Google types and Foursquare categories, we can determine the wOndary categorization. The Colosseum would be categorized into *Culture* because the Google types (*'point_of_interest'* and *'establishment'*) are not part of the mapping, whereas a *'Historic Site'* maps to the *Culture* category. One attraction can have several wOndary categories; however, not all venue types are relevant for travelers. For example, hospitals are not mapped to any of our categories, because we argue that they are not relevant for planning a trip.

Figure 2 shows the distribution of categories of a representative sample of 150 trips from wOndary based on the top trips according to user interactions. A closer look into the distribution of the categories in Figure 3 shows that most venues are categorized into the *Outdoor* category, whereas *Nightlife* is the least frequent.

Having classified the items, it is also necessary to know the user's preferences to do content-based recommendation. The default method would be to explicitly ask the user to indicate her preferences regarding the five travel categories, e.g., on a scale from 1 (not interesting) to 5 (highly interesting). However, this would require a manual interaction, which we can avoid by using synergy effects from the categorization of attractions. To answer RQ3, we aggregated all attractions from a user's saved trips to create a user preference profile. While this can be refined further with more detailed click stream data, it is a straightforward metric for classifying user travel preferences within wOndary.

The actual ranking for the recommendations is performed by calculating the cosine similarity using the five dimensional vector of distinctive travel interests. Here we exploit the structure of our data model to recommend complete trips and trip parts, i.e., blocks or specific days. Currently, the Explore page features trips and blocks as recommendations. In the first step, the system filters out all trips that are not within the bounds of the map or do not match the temporal filters (see Figure 1). When a trip is only partially in the query region, the blocks within the area will be included. Then, all past trips of the user are removed because we assume they are not of interest for future travel plans. Trips and blocks as ranked by the cosine similarity with respect to the user profile and also listed left of the map. To keep clarity in the interface, only the top 30 items are displayed.

For new users that have not yet copied any trips, the content-based recommender cannot compute a ranking for the trips. Therefore, the trips displayed on the Explore page will be ranked by the geographic distance to the center of the map.

[2]https://developers.google.com/places/web-service/supported_types
[3]https://developer.foursquare.com/docs/resources/categories

3 STATE OF THE ART OF PERSONALIZED TRAVEL RECOMMENDATION

The tourism domain is a popular branch of recommender system (RS) research because it is a highly emotional, personal, and inherently complex topic. Early systems recommended single items, such as attractions or bundled travel packages [17], and there are big commercial players, such as hotels, restaurants, airlines, and activities. In their survey, Borràs et al. [3] categorized an intelligent tourism RS into four functionalities: travel destination and tourist packs, suggested attractions, trip planners, and social aspects. In 2014, most approaches focused on the attraction suggestion category; however, currently, the trend is on complex recommendations [30], such as sequences of attractions [29], composite travel regions [6, 13], and group recommendations [5] for tourism. When it comes to complex recommendations such as enjoyable routes, the challenge is to identify relevant points of interest (POIs) and then connect them in a coherent trip. This problem is called the Tourist Trip Design Problem (TTDP) [9], which is algorithmically interesting and has been widely investigated [28].

However, in this paper, we tackled the complexities of travel using a crowdsourcing approach by performing personalized travel recommendations using actual trips from users. Crowdsourcing has the advantage of being able to vary the length of travel, such as a multi-month world trip, a week trip to an island, or a weekend in a city, and this is an unsolved challenge in the tourist RS for solving the TTDP. Furthermore, the structured representation of trips allows the combination of several independent blocks into a prolonged trip or the possibility of selecting parts of a trip if the traveler is short on time. Determining the duration of stay at each location can be further personalized with additional information about the traveler, such as tourist mobility patterns [7] from past trips.

The RS of static travel items utilizes ratings as one factor of a hybrid recommendation algorithm [4]. However, because we exploit the trip structures to aggregate and reassemble trips, ratings are not of much use due to their high sparsity. Furthermore, we are concerned that users are not motivated to provide ratings for trips and blocks, and the platform's user experience could decline if it required users to rate trips. Therefore, we have employed the content-based recommendation paradigm [23] to match items to users. Content-based recommendations are commonplace as a hybrid factor in complex domains, such as in scientific publications [1], news articles [15, 16] or tourism [14]. However, for a purely content-based recommendation, it is often challenging to model the user after the very same features as the items to compute a similarity measure, e.g., the cosine distance, for ranking items. When investigating potential classification schemes of touristic items for content-based recommendations, the work of Neidhardt et al. is an established alternative to wOndary's categorization. Based on the Big Five Factor Model [18] from personality psychology and prior research on tourist roles [10, 31], Neidhardt et al. developed the *Seven Factor Model* of tourist behavioral patterns [21]. In a follow-up study [22], they showed that this can be used to elicit user preferences via pictures classified by domain experts. However, the final step of using these tourist behavioral patterns to recommend items was only recently performed [25] and required a very big commercial data set of 30,000 tourist destinations classified along 27 motivational and 14 geographical attributes.

Commercial approaches for travel recommendations range from merchants focusing on the sale of travel-related services, such as activities, transport and lodging, to review platforms with a business model based on commissions. Depending on the type of business, travel recommendations are a side-product or a main feature in which the recommendation can include a single product or service or complete trips. Big platforms, such as TripAdvisor and Google Maps, recommend separate activities to users based on ratings, reviews, and behavior on the platform. Social networks, such as Facebook, provide less structured ways to ask friends for travel recommendations as a way to provide crowd-sourcing recommendations. Google Trips recommends single- or multi-day tours [8] in the vicinity based on user behavior and by scanning the user's booking confirmations in Gmail.

Mafengwo[4] and Qyer[5] (both solely available in Mandarin) are the closest platforms to our approach and provide travel-related services, as well as trip planning, and sharing functionalities.

4 AUGMENTING WONDARY'S TRAVEL RECOMMENDATIONS

As described in this paper, its core functionality is the first step in wOndary's travel RS. To answer the second part of RQ1, this section discusses wOndary's future agenda concerning trip recommendations. We plan to improve the item categorization, to enable explanations and critiquing of our recommendations, and to explicitly support the travel decision-making process for groups.

As discussed at the end of Section 2, currently, new users are not provided with content-based recommendations. We believe we can overcome the cold start problem with an elaborate click stream analysis and an initial preference elicitation phase in which users provide their feedback for the five categories e.g., through small games.

The current categorization is based on expert knowledge and data sources for categorization. It would be useful to do a thorough investigation of the attraction's attributes with unsupervised learning to obtain data-backed clusters. Furthermore, a latent factor analysis of the trips would be interesting to evaluate the explicit categories. As we have rich information about the trips, the core of our recommender system is content-based. This could be improved in the future with more hybrid factors, e.g., knowledge-based recommendations and collaborative ratings of items. To provide transparency and improve trust in the recommendations, it would be highly interesting to provide explanations of the recommendations [27] to the user. These explanations could be based on the classification of items (*"because you liked ..."*), the users (*"travelers similar to you also liked ..."*), or by taking the social network on the platform into account (*"your friend traveled to ..."*) [2]. Another promising technique to improve recommendations is critiquing [19]. A conversational element [20] within the presentation of results would enable active learning of user needs [24]. This is useful because we think that it is unlikely that recommendations in such a complex domain are perfect on the first iteration, e.g., because

[4] http://www.mafengwo.cn/
[5] http://www.qyer.com/

travelers may want to go on a different type of holiday than they went on before.

Since travel planning on wOndary is already collaborative, it is a logical step to extend the recommendations to groups to support the decision-making process. However, we acknowledge that this issue has not been resolved and is still of high interest to the research community in this area [5, 12, 26].

5 CONCLUSIONS

In this paper, we described our approach for recommending high-quality crowdsourced trips. We presented a novel structure for itineraries that is both useful for users to obtain an understanding of trip characteristics and for software systems to work with. This data model structures user-submitted trips, thereby defining items based on different lengths, i.e., trips, blocks, and days. Second, we have automatically classified the aforementioned item types using wOndary's categorization scheme. We exploited this categorization scheme to perform user modeling without explicit elicitation of preferences using the user's past trips. Finally, we showcased a user interface for intuitively presenting recommendations for trips across the globe.

The current version is the core part of the content-based recommender system and will be extended with advanced features in the future. While the recommendations of this platform are more personalized than ranking trips by distance to the center of the map, we want to confirm the perceived accuracy by utilizing an automated A/B testing framework, which we will also use to continuously measure future changes in the algorithm, of which we have sketched several in our future work section. This will establish wOndary's test setup for travel recommendations to provide informed decisions regarding improvement of the product.

REFERENCES

[1] Joeran Beel, Akiko Aizawa, Corinna Breitinger, and Bela Gipp. 2017. Mr. DLib: Recommendations-as-a-Service (RaaS) for Academia. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL'17)*. IEEE, 1–2. https://doi.org/10.1109/JCDL.2017.7991606

[2] Philip Bonhard and M. Angela Sasse. 2006. 'Knowing me, knowing you' — Using profiles and social networking to improve recommender systems. *BT Technology Journal* 24, 3 (July 2006), 84–98. https://doi.org/10.1007/s10550-006-0080-3

[3] Joan Borràs, Antonio Moreno, and Aida Valls. 2014. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications* 41, 16 (Nov. 2014), 7370–7389. https://doi.org/10.1016/j.eswa.2014.06.007

[4] J. H. Chen, K. M. Chao, and N. Shah. 2013. Hybrid Recommendation System for Tourism. In *2013 IEEE 10th International Conference on e-Business Engineering*. IEEE, 156–161. https://doi.org/10.1109/icebe.2013.24

[5] Amra Delic, Judith Masthoff, Julia Neidhardt, and Hannes Werthner. 2018. How to Use Social Relationships in Group Recommenders: Empirical Evidence. In *26th Conference on User Modeling, Adaptation and Personalization (UMAP '18)*. ACM, New York, NY, USA, 121–129. https://doi.org/10.1145/3209219.3209226

[6] Linus W. Dietz. 2018. Data-Driven Destination Recommender Systems. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP '18)*. ACM, New York, NY, USA. https://doi.org/10.1145/3209219.3213591

[7] Linus W. Dietz, Daniel Herzog, and Wolfgang Wörndl. 2018. Deriving Tourist Mobility Patterns from Check-in Data. In *Proceedings of the WSDM 2018 Workshop on Learning from User Interactions*. Los Angeles, CA, USA.

[8] Zachary Friggstad, Sreenivas Gollapudi, Kostas Kollias, Tamas Sarlos, Chaitanya Swamy, and Andrew Tomkins. 2018. Orienteering Algorithms for Generating Travel Itineraries. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 180–188. https://doi.org/10.1145/3159652.3159697

[9] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou. 2014. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics* 20, 3 (June 2014), 291–328. https://doi.org/10.1007/s10732-014-9242-5

[10] Heather Gibson and Andrew Yiannakis. 2002. Tourist Roles: Needs and the Lifecourse. *Annals of Tourism Research* 29, 2 (April 2002), 358–383.

[11] Google Consumer Insights. 2016. Travel Micro-Moments Guide. Online. (July 2016). https://www.thinkwithgoogle.com/marketing-resources/micro-moments/micro-moments-reshaping-travel-customer-journey/

[12] Daniel Herzog. 2017. Recommending a Sequence of Points of Interest to a Group of Users in a Mobile Context. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 402–406. https://doi.org/10.1145/3109859.3109860

[13] Daniel Herzog and Wolfgang Wörndl. 2014. A Travel Recommender System for Combining Multiple Travel Regions to a Composite Trip. In *CBRecSys@RecSys (CEUR Workshop Proceedings)*, Vol. 1245. CEUR-WS.org, Foster City, Silicon Valley, California, USA, 42–48.

[14] Daniel Herzog and Wolfgang Wörndl. 2016. Extending Content-Boosted Collaborative Filtering for Context-aware, Mobile Event Recommendations. In *Proceedings of the 12th International Conference on Web Information Systems and Technologies*. SCITEPRESS - Science and and Technology Publications, 293–303. https://doi.org/10.5220/0005763702930303

[15] Philip Lenhart and Daniel Herzog. 2016. Combining Content-based and Collaborative Filtering for Personalized Sports News Recommendations. In *CBRecSys@RecSys (CEUR Workshop Proceedings)*, Vol. 1673. CEUR-WS.org, 3–10.

[16] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized News Recommendation Based on Click Behavior. In *15th International Conference on Intelligent User Interfaces (IUI'10)*. ACM. https://doi.org/10.1145/1719970.1719976

[17] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. 2011. Personalized Travel Package Recommendation. In *Proceedings of the IEEE 11th International Conference on Data Mining (ICDM '11)*. IEEE, Vancouver, BC, Canada, 407–416. https://doi.org/10.1109/icdm.2011.118

[18] Robert R. McCrae and Oliver P. John. 1992. An Introduction to the Five-Factor Model and Its Applications. *Journal of Personality* 60, 2 (June 1992), 175–215. Issue 2. https://doi.org/10.1111/j.1467-6494.1992.tb00970.x

[19] Lorraine McGinty and James Reilly. 2011. On the Evolution of Critiquing Recommenders. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer US, Boston, MA, USA, 419–453. https://doi.org/10.1007/978-0-387-85820-3_13

[20] Lorraine McGinty and Barry Smyth. 2006. Adaptive Selection: An Analysis of Critiquing and Preference-Based Feedback in Conversational Recommender Systems. *International Journal of Electronic Commerce* 11, 2 (Dec. 2006), 35–57. https://doi.org/10.2753/jec1086-4415110202

[21] Julia Neidhardt, Rainer Schuster, Leonhard Seyfang, and Hannes Werthner. 2014. Eliciting the Users' Unknown Preferences. In *Proceedings of the 8th ACM Conference on Recommender systems (RecSys '14)*. ACM, New York, NY, USA, 309–312. https://doi.org/10.1145/2645710.2645767

[22] Julia Neidhardt, Leonhard Seyfang, Rainer Schuster, and Hannes Werthner. 2015. A picture-based approach to recommender systems. *Information Technology & Tourism* 15, 1 (March 2015), 49–69. https://doi.org/10.1007/s40558-014-0017-5

[23] Michael J. Pazzani and Daniel Billsus. 2007. *Content-Based Recommendation Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 325–341. https://doi.org/10.1007/978-3-540-72079-9_10

[24] Neil Rubens, Dain Kaplan, and Masashi Sugiyama. 2015. Active Learning in Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, Boston, MA, USA, 809–846. https://doi.org/10.1007/978-1-4899-7637-6_24

[25] Mete Sertkan, Julia Neidhardt, and Hannes Werthner. 2018. Mapping of Tourism Destinations to Travel Behavioural Patterns. In *Information and Communication Technologies in Tourism 2018*, Brigitte Stangl and Juho Pesonen (Eds.). Springer International Publishing, Cham, 422–434. https://doi.org/10.1007/978-3-319-72923-7_32

[26] Kadri Sylejmani, Jürgen Dorn, and Nysret Musliu. 2017. Planning the trip itinerary for tourist groups. *Information Technology & Tourism* 17, 3 (Sept. 2017), 275–314. https://doi.org/10.1007/s40558-017-0080-9

[27] Nava Tintarev and Judith Masthoff. 2007. A Survey of Explanations in Recommender Systems. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*. IEEE, 801–810. https://doi.org/10.1109/ICDEW.2007.4401070

[28] Pieter Vansteenwegen and Dirk Van Oudheusden. 2007. The Mobile Tourist Guide: An OR Opportunity. *OR Insight* 20, 3 (July 2007), 21–27. https://doi.org/10.1057/ori.2007.17

[29] Wolfgang Wörndl, Alexander Hefele, and Daniel Herzog. 2017. Recommending a sequence of interesting places for tourist trips. *Information Technology & Tourism* 17, 1 (Feb. 2017), 31–54. https://doi.org/10.1007/s40558-017-0076-5

[30] Min Xie, Laks V.S. Lakshmanan, and Peter T. Wood. 2010. Breaking out of the Box of Recommendations: From Items to Packages. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 151–158. https://doi.org/10.1145/1864708.1864739

[31] Andrew Yiannakis and Heather Gibson. 1992. Roles tourists play. *Annals of Tourism Research* 19, 2 (jan 1992), 287–303. https://doi.org/10.1016/0160-7383(92)90082-z

Integrating Public Displays into Tourist Trip Recommender Systems

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

Nikolaos Promponas-Kefalas
Department of Informatics
Technical University of Munich
85748 Garching bei München,
Germany
nikolas.promponas@tum.de

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

ABSTRACT

Tourist Trip Recommender Systems (RSs) suggest points of interest (POIs) and combine them along enjoyable routes. Integrating public displays into the recommendation process promises to overcome the limitations of mobile devices, such as small screens, thereby enriching the user experience of a tourist trip RS. However, in practice, public displays are rarely integrated in this manner. In this paper, we show how a mobile RS for tourist trips can be adapted to public displays and propose a Distributed User Interface (DUI) approach where the RS is distributed among both public and private devices. The results of a preliminary user study indicate that integrating public displays is perceived as attractive and novel; however, people remain concerned about privacy issues when using a public display. Public displays become more interesting when used for group recommendations; thus, we outline how our proposed approaches can be integrated into a group RS.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Human-centered computing → Interaction techniques**; *Touch screens*;

KEYWORDS

Tourist Trip, Smartphone, Public Display, Distributed User Interface, Usability

1 INTRODUCTION

Recommender Systems (RSs) have been applied in various domains; however they are particularly popular in tourism where they allow users to receive suggestions for points of interest (POIs) or tourist trips comprising multiple attractions [2]. Today, mobile devices are the primary information access platform and tourists use mobile RSs to receive recommendations when traveling [18]. However, mobile RSs have to deal with various limitations, such as small displays and limited Internet access.

Public displays at touristic spots are one solution to overcome these limitations. Currently, public displays are used to display primarily static content, such as maps and timetables. The next step in public display research is to provide personalized content tailored to individual information needs by allowing the user to interact with the display and share their preferences. However, using public displays for personalized recommendations raises privacy issues, and some people are reluctant to use them because of social embarrassment [4]. A Distributed User Interface (DUI) is one solution to this problem. With a DUI, the user interacts solely with their personal device to protect sensitive data, and the public display only receives and displays selected content, such as the final recommendation.

Our overall goal is to integrate public displays into tourism RSs. Previously, we developed TourRec, a tourist trip RSs that recommends sequences of POIs along enjoyable routes between a start point and a destination and adapts the routes according to user preferences and constraints [9]. An updated version of TourRec is available on the Google Play Store[1]. In this paper, we show how TourRec can be adapted to public displays and introduce a DUI approach that combines the advantages of public and private devices. In a preliminary user study, we evaluated different approaches relative to usability criteria. In addition, we describe how to integrate our approaches into a group RSs.

The remainder of this paper is organized as follows: In Section 2 we present background information and related work. We introduce our approaches to integrate public displays into a tourist trip RSs and the results of a preliminary user study in Section 3. In addition, we explain how our ideas can be used in group RSs. Conclusions and suggestions for future work are given in Section 4.

2 BACKGROUND AND RELATED WORK

Tourism RSs can recommend different travel-related items, such as POIs, travel plans, and tourist trips [2]. Traditional tourism RSs often recommend sets or ranked lists of POIs from which the user can choose the attractions they would like to visit. The recommendations can be optimized by considering contextual factors, such as weather [3]. More advanced RSs can recommend complete travel plans composed of multiple travel items, such as a destination, a hotel, and nearby POIs. TripMatcher and VacationCoach are early travel RSs that use content-based approaches to match user preferences with potential destinations [17]. Other approaches to generate travel plans involve case-based reasoning [11, 19] or conversational UIs [12]. However, tourist trips are sequences of POIs along enjoyable routes [23]. For example, the City Trip Planner is a tourist trip RS that generates personalized routes and can integrate lunch breaks [22]. Another example is TourRec, the mobile RS that is the basis of this work (Section 3.1). Other tourist trip RSs identify routes that are considered scenic or pleasant [6, 16]. Only very few work has been done to recommend tourist trips to groups [20].

In the tourism domain, RSs are typically developed for mobile devices or desktop clients. Another idea to provide personalized recommendations to people who are already traveling is deployment

[1] https://play.google.com/store/apps/details?id=de.tum.in.cm.tourrec

on public displays, such as information kiosks in trains stations, airports, and touristic areas. Public displays vary in size from small television screens to display static information, such as timetables, to large and interactive multi-user wall displays [15]. Interactive public displays can be differentiated based on input types and interaction techniques. Users can interact directly with a touchscreen or buttons attached to the display or they can use speech or gestures [14].

Even though public displays have many advantages compared to mobile devices, such as screen size, social embarrassment and privacy concerns prevent people from interacting with them. People may be uncomfortable entering sensitive data in a publicly available device [4]. Furthermore, passersby may be able to see sensitive content, a phenomenon referred to as *shoulder-surfing* [5]. Using a mobile device to enter personal information is a promising way to address privacy issues [1]. UIs that are distributed across multiple devices or interfaces are referred to as DUIs. With DUIs, the UIs can be displayed on different monitors, devices, or platforms, and they can be distributed among different users [21].

3 INTEGRATING PUBLIC DISPLAYS INTO A TOURIST TRIP RECOMMENDER SYSTEM

In this section, we describe TourRec and show how it can be adapted to public displays. We also introduce a DUI approach that combines private and public devices. In addition, we summarize the results of a preliminary user study performed to evaluate these approaches and explain how such approaches can be integrated into a group RSs.

3.1 TourRec

TourRec is a mobile tourist trip RS that combines multiple POIs along enjoyable routes [9]. Prior to requesting a recommendation, the user can specify their travel preferences by rating various categories, such as *food* or *nightlife*, on a scale from 0 to 5. The user must specify an origin (e.g., the user's current location), a destination, a start time, and the maximum duration of the trip. A *PlacePicker* UI allows to select the origin and destination by searching for a location or selecting it directly on a map. Recommendations are displayed on a map or as a list of POIs with additional information, such as predicted arrival times and suggested durations of stay (Figure 1a). Note that the UIs were designed using Material Design, a design language introduced by Google[2].

To generate tourist trips based on user queries, TourRec communicates with a backend we developed for this purpose. The backend architecture is modular and scalable, which allows us to add and evaluate new clients, recommendation algorithms, and data sources.

3.2 Public Display Variant

Public displays have become increasingly common in touristic areas; however, they are still not used for personalized recommendations. The potential advantages are obvious. The user does not need their own device with Internet connection while traveling. Larger displays can facilitate orientation in an unknown area and support the selection of a suitable recommendation when all relevant data, such as POI information, a map, and context data, are displayed on a single UI. Furthermore, a public display can facilitate decision making when used by a group because the recommendation can be viewed by all members of the group. More advanced approaches allow the user to modify the recommendation directly on the public display and send it to the personal device. These advantages represent our motivation to integrate public displays into our TourRec application.

In this work, we use a kiosk system equipped with a 55-inch multi-touch screen in portrait orientation. Similar tourism information kiosks can be found in many touristic areas. We tried to keep the changes to the smartphone's UIs to a minimum so that the only independent variable tested in our user study was the interaction type rather than other changes in the layout. Thus, the public display application applies the same layout but attempts to benefit from the larger display area wherever possible.

Figure 1b shows a tourist trip recommendation on the public display. Again, the final recommendation is presented both on a map and as a list of POIs. However, the public display variant takes advantage of the larger screen and displays both modes simultaneously. The map and list are displayed on the top and bottom of the screen, respectively.

We used the AngularJS framework to implement the public display application. The kiosk system runs Windows 10 and the application can be accessed via any web browser.

3.3 Distributed User Interface Approach

The DUI approach distributes the recommendation process among the smartphone and the public display. The two main reasons for this approach are: (i) users can keep sensitive data on their private device but view the recommendation on a large display, and (ii) users can prepare a route request prior to traveling and display a recommendation on a public display as required.

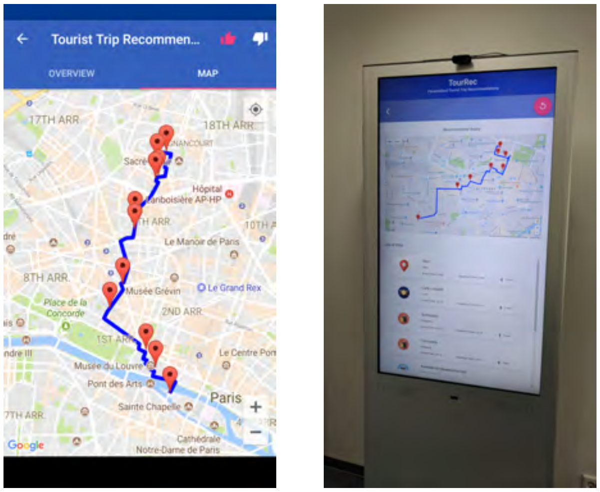
We decided to use a QR code for the pairing between the smartphone and public display because it has been shown that this method provides high usability in similar scenarios [24]. Furthermore, QR codes are already used in common software, such as WhatsApp[3], to pair a desktop client and a smartphone.

After the user formulates a route request, the extended smartphone application allows the user to send the recommendation to a public display. The user must scan the QR code using the smartphone's camera to transmit the request to an intermediary server application we have developed. The public display fetches the route request from the intermediary server application. To identify the correct smartphone, each request is labeled with a unique ID that is also encoded in the QR code. After the public display receives the request, it forwards it to the backend and receives a recommendation, which is then presented to the user.

The smartphone and public display applications are the same as in Sections 3.1 and 3.2; however, they are extended by the pairing feature. The intermediary server application is a web service implemented in node.js.

[2] https://material.io/guidelines/

[3] https://www.whatsapp.com/



(a) (b)

Figure 1: Tourist trip recommendation in a) the mobile application and b) the public display application

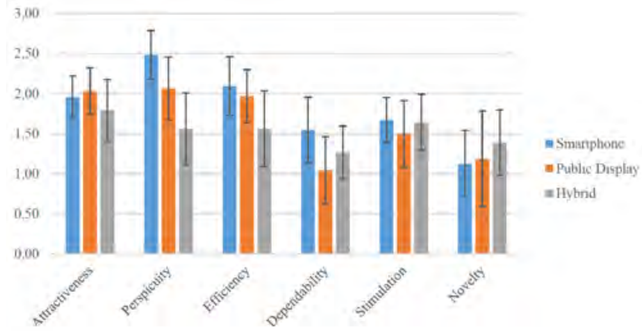


Figure 2: UEQ results for three interaction techniques

Table 1: Task Times

	Task 1	Task 2	Task 3
Smartphone	33.38 s	77.44 s	25.56 s
Public Display	34.69 s	73.06 s	33.63 s
DUI approach	43.81 s	81.19 s	36.81 s

3.4 Preliminary User Study

We conducted a preliminary user study to obtain initial feedback on the proposed integrations of public displays.

3.4.1 Goals and Setup. We evaluated the three variants of the single-user RS relative to user experience, execution time of the selected tasks, and comfortability of use in a public space. The user study followed a within-group design. We allowed the participants to test the prototypes in random order to avoid biased results due to the learning effect. The participants were asked to execute three tasks for each interaction technique: (i) create a route between two predefined POIs, (ii) create a route between two predefined POIs with their own travel preferences, and (iii) create a route from the current location to a predefined destination.

The participants were asked to fill out a User Experience Questionnaire (UEQ) after every interaction technique. The UEQ is a semantic differential with 26 items grouped into six user experience aspects: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty [10]. A benchmark data set that enables comparison of the performance of each aspect to other systems, exists. In addition, we included one extra question asking the user how comfortable they felt using the prototype in a public place.

In total, 16 people participated in the user study. All participants were bachelor or master's degree students or had recently graduated. Overall, the participants had rather limited experience with interactive public displays, e.g., 50% of the participants had never used a similar system previously.

3.4.2 Results and Discussion. We performed statistical tests where applicable to determine whether the performance of the interaction techniques differed significantly at the 5% level relative to any of the aforementioned aspects. We used Repeated Measures Analysis of Variance (ANOVA) when the results were distributed normally and the Friedman test in other cases. The Shapiro-Wilk test for normality was performed to select the correct significance test. In case of a significant difference, we performed a post-hoc test to identify where the difference occurred, i.e., between interaction techniques.

Figure 2 shows the results of all prototypes relative to the six UEQ aspects. As can be seen, the attractiveness of all prototypes is considered *excellent*, which means that it is among the 10% best results of the benchmark data set. However, perspicuity is significantly higher for the stand-alone smartphone mode compared to the DUI approach ($\alpha = 0.002$). Many people are familiar with using smartphone applications. Hence, it is easier for them to get familiar with the stand-alone smartphone variant than a hybrid approach. For dependability, the difference between the stand-alone smartphone and public display modes is significant ($\alpha = 0.006$), which means that the participants felt more in control of the interaction when using a smartphone than a public display. Moreover, the public display's dependability score was *below average* compared to the benchmark dataset because the public display scored very low for the *Secure vs Insecure* item. Thus, further effort to protect user data and prevent *shoulder-surfing* is required. Our DUI approach appears to be a promising solution because its dependability is similar to the stand-alone smartphone variant. Furthermore, the DUI approach demonstrates the highest novelty, which means that this approach feels the most innovative and creative. However, this difference is not yet significant.

Table 1 shows the average execution times for each task and prototype. The execution times of Task 1 are significantly shorter for both the stand-alone smartphone mode ($\alpha = 0.007$) and the stand-alone public display mode ($\alpha = 0.015$) than the DUI approach. Task 3, which requires the user to give the system access to their current location, is significantly faster on the smartphone than on the public display ($\alpha = 0.002$) and for the DUI approach ($\alpha = 0.003$). There is no significant difference between the execution times of Task 2 which included entering the travel preferences before requesting a recommendation.

The analysis of execution time shows that there is nearly no difference between the public display and smartphone interaction techniques. This is surprising because many participants had no previous experience with interactive public displays.

Comfortability using a smartphone in a public place is significantly higher than when using a public display ($\alpha = 0.005$) and using the DUI approach ($\alpha = 0.005$). During the study, 75% of participants explained that using two devices is a disadvantage and too complex because they could obtain the same recommendation using a single device. However, 25% of participants emphasized that preparing the route recommendation in advance, e.g., by entering route parameters on the smartphone, while waiting in line to use the public display could be a significant advantage in practical use.

3.5 Group Recommender System

The results of the preliminary user study show that integration of public displays into a tourist trip RS is perceived as attractive and novel. However, the advantages of a hybrid approach are less appreciated when used by single users. The feedback received indicates that public displays could become more valuable when a group of users attempts to agree on a tourist trip.

The simplest ways to find a group recommendation is to use only a single smartphone or allow one group member to use the public display on behalf of the group; however, this requires the group members to agree on the group's travel preferences in advance. A more sophisticated approach uses one smartphone per user, thereby allowing each user to independently enter their travel preferences. In this case, the preferences of all users are merged automatically by the RS using a social choice strategy [13]. Networking Application Programming Interfaces (APIs), such as Google Nearby[4], can be used to share travel preferences and recommendations between smartphones without an Internet connection. Thus, only one device is required to request a recommendation from the backend and broadcast it to the other users. One advantage of this approach is that the users do not have to reveal their travel preferences, which avoids social embarrassment and manipulation [7].

When a public display is available, no Internet-connected device is required. The public display variant presented in Section 3.2 can be used by a group if the group's preferences are entered by a single group member. Furthermore, we suggest an extension to our DUI approach where each user enters their preferences using their personal device and the recommendation is displayed on a mutual screen. Thus, this approach uses the same UIs as the previous prototypes. However, the collected preferences are aggregated automatically before the public display shows the recommendation to the group. This approach combines different advantages of the previous solutions, i.e., users do not have to reveal travel preferences and the mutual display facilitates discussions among group members, which helps the group determine consensus [8]. More advanced approaches allow the group to modify the recommendation on the public display and send it back to their devices.

To summarize, the strategies we suggest for a group RS can be distinguished by the following dimensions.

- Small screen vs. large screen
- One user enters the preferences for the group vs. every user enters the preferences separately
- User preferences are revealed to the group vs. preferences are hidden from the group
- The recommendation is displayed on a mutual screen vs. the recommendation is displayed on each individual's device

Our goal is to compare these different approaches to determine which specifications under which conditions facilitate the process of finding a tourist trip for a group. For this purpose, we plan to conduct user studies with different group types, such as families, friends, or colleagues. The results will show us which approaches perform best relative to different usability criteria and if there are any differences depending on the type of user and group.

4 CONCLUSION AND FUTURE WORK

In this paper, we have shown how public displays can be integrated into a tourist trip RS. We adapted a smartphone application to public displays and extended it with a pairing functionality to realize a DUI approach. In a preliminary user study, very high attractiveness was demonstrated by all approaches. However, the results of our preliminary study show that public displays provide limited advantage when used by single users. As a result, we have outlined how our approaches can be extended to enable group recommendations. In future, we will implement these extensions and compare the different approaches in larger user studies with real groups to evaluate how they support tourist trip recommendations in a group context.

Integrating public displays into a tourist trip RS offers tourists many advantages compared to mobile devices; however, privacy concerns relative to using a public display remain. The DUI approach presented in this work allows the user to keep sensitive data, such as travel preferences, on the private device while benefiting from the public display. This is particularly important in a group recommendation scenario where the users want to share a mutual display but not reveal their personal preferences to other group members. However, further efforts are needed to protect the data on the public display, such as the actual recommendation. Different approaches to prevent *shoulder-surfing*, such as blacking out parts of the display or mirroring a passerby's position and orientation to warn the user have been developed [5]. Future work should evaluate how these approaches can be adapted to the tourist trip scenario and group recommendations on public displays.

One main limitation of our preliminary study is the fact that it was conducted as a lab study. The advantages of a large display could become clearer when the user is actually traveling. On the other hand, social embarrassment and privacy concerns could become a significantly larger issue when passersby are present. Hence, in a real-world scenario, the users could accept the presented DUI approach more easily compared to the public display variant.

ACKNOWLEDGMENTS

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bavarian Ministry of Economic Affairs, Energy and Technology (StMWi) through the Center Digitisation.Bavaria, an initiative of the Bavarian State Government.

[4]https://developers.google.com/nearby/

REFERENCES

[1] Florian Alt, Alireza Sahami Shirazi, Thomas Kubitza, and Albrecht Schmidt. 2013. Interaction Techniques for Creating and Exchanging Content with Public Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1709–1718. https://doi.org/10.1145/2470654.2466226

[2] Joan Borràs, Antonio Moreno, and Aida Valls. 2014. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications* 41, 16 (2014), 7370 – 7389. https://doi.org/10.1016/j.eswa.2014.06.007

[3] Matthias Braunhofer, Mehdi Elahi, Mouzhi Ge, and Francesco Ricci. 2013. STS: Design of Weather-Aware Mobile Recommender Systems in Tourism. In *In Proceedings of the 1st Workshop on AI*HCI: Intelligent User Interfaces (AI*HCI 2013)*.

[4] Harry Brignull and Yvonne Rogers. 2003. Enticing People to Interact with Large Public Displays in Public Spaces. In *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*.

[5] Frederik Brudy, David Ledo, Saul Greenberg, and Andreas Butz. 2014. Is Anyone Looking? Mitigating Shoulder Surfing on Public Displays Through Awareness and Protection. In *Proceedings of The International Symposium on Pervasive Displays (PerDis '14)*. ACM, New York, NY, USA, Article 1, 6 pages. https://doi.org/10.1145/2611009.2611028

[6] Damianos Gavalas, Vlasios Kasapakis, Charalampos Konstantopoulos, Grammati Pantziou, and Nikolaos Vathis. 2016. Scenic route planning for tourists. *Personal and Ubiquitous Computing* (2016), 1–19. https://doi.org/10.1007/s00779-016-0971-3

[7] Anthony Jameson. 2004. More Than the Sum of Its Members: Challenges for Group Recommender Systems. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 48–54. https://doi.org/10.1145/989863.989869

[8] Anthony Jameson and Barry Smyth. 2007. Recommendation to Groups. In *The Adaptive Web*, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Springer, Berlin, Heidelberg, 596–627. http://dl.acm.org/citation.cfm?id=1768197.1768221

[9] Christopher Laß, Wolfgang Wörndl, and Daniel Herzog. 2016. A Multi-Tier Web Service and Mobile Client for City Trip Recommendations. In *The 8th EAI International Conference on Mobile Computing, Applications and Services (MobiCASE)*. ACM. https://doi.org/10.4108/eai.30-11-2016.2267194

[10] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *HCI and Usability for Education and Work*, Andreas Holzinger (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 63–76.

[11] Mario Lenz. 1994. CaBaTa: Case-based Reasoning for Holiday Planning. In *Proceedings of the International Conference on Information and Communications Technologies in Tourism*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 126–132. http://dl.acm.org/citation.cfm?id=184620.184769

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

[13] Judith Masthoff. 2004. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Modeling and User-Adapted Interaction* 14, 1 (Feb. 2004), 37–85. https://doi.org/10.1023/B:USER.0000010138.79319.fd

[14] Jörg Müller, Florian Alt, Daniel Michelis, and Albrecht Schmidt. 2010. Requirements and Design Space for Interactive Public Displays. In *Proceedings of the 18th ACM International Conference on Multimedia (MM '10)*. ACM, New York, NY, USA, 1285–1294. https://doi.org/10.1145/1873951.1874203

[15] Peter Peltonen, Esko Kurvinen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, John Evans, Antti Oulasvirta, and Petri Saarikko. 2008. It's Mine, Don'T Touch!: Interactions at a Large Multi-touch Display in a City Centre. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1285–1294. https://doi.org/10.1145/1357054.1357255

[16] Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. 2014. The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy Routes in the City. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media (HT '14)*. ACM, New York, NY, USA, 116–125. https://doi.org/10.1145/2631775.2631799

[17] Francesco Ricci. 2002. Travel Recommender Systems. *IEEE Intelligent Systems* (2002), 55–57.

[18] Francesco Ricci. 2010. Mobile Recommender Systems. *Information Technology & Tourism* 3 (2010), 205–231. https://doi.org/doi:10.3727/109830511X12978702284390

[19] F. Ricci, D. R. Fesenmaier, N. Mirzadeh, H. Rumetshofer, E. Schaumlechner, A. Venturini, K. W. Wöber, and A. H. Zins. 2006. DieToRecs: a case-based travel advisory system. In *Destination recommendation systems: behavioural foundations and applications*, D. R. Fesenmaier, K. W. WÃűber, and H. Werthner (Eds.). CABI, 227–239.

[20] Kadri Sylejmani, Jürgen Dorn, and Nysret Musliu. 2017. Planning the trip itinerary for tourist groups. *Information Technology & Tourism* 17, 3 (01 Sep 2017), 275–314. https://doi.org/10.1007/s40558-017-0080-9

[21] Jean Vanderdonckt. 2010. Distributed User Interfaces: How to Distribute User InterfaceElements across Users, Platforms, and Environments. In *Proc. of XIth Congreso Internacional de Interacción Persona-Ordenador Interacción'2010*. Valencia, 3–14.

[22] Pieter Vansteenwegen, Wouter Souffriau, Greet Vanden Berghe, and Dirk Van Oudheusden. 2011. The City Trip Planner. *Expert Syst. Appl.* 38, 6 (June 2011), 6540–6546. https://doi.org/10.1016/j.eswa.2010.11.085

[23] Pieter Vansteenwegen and Dirk Van Oudheusden. 2007. The mobile tourist guide: an OR opportunity. *OR Insight* 20, 3 (2007), 21–27.

[24] Jouni Vepsäläinen, Antonella Di Rienzo, Matti Nelimarkka, Jouni A. Ojala, Petri Savolainen, Kai Kuikkaniemi, Sasu Tarkoma, and Giulio Jacucci. 2015. Personal Device As a Controller for Interactive Surfaces: Usability and Utility of Different Connection Methods. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 201–204. https://doi.org/10.1145/2817721.2817745

Roadscape-based Route Recommender System using Coarse-to-fine Route Search

Koji Kawamata
Ryukoku University
Otsu, Shiga, Japan
t18m057@mail.ryukoku.ac.jp

Kenta Oku
Ryukoku University
Otsu, Shiga, Japan
okukenta@rins.ryukoku.ac.jp

ABSTRACT

We propose the Roadscape-based Route Recommender System (R3), which provides diversified roadscape-based routes. Given starting and destination points, R3 provides four types of roadscape-based routes: rural-, mountainous-, waterside-, and urban-prior routes. To reduce the computational cost, we propose a coarse-to-fine route search approach that consists of a roadscape-based clustering method, a roadscape cluster graph, a coarse-grained route search, and a fine-grained route search. We evaluated the performance of R3 using network data for a real road. The experimental results show that using coarse-grained route search can significantly reduce route search time.

CCS CONCEPTS

• **Information systems → Social recommendation**;

KEYWORDS

route recommender system, route search, roadscape

1 INTRODUCTION

Cars are driven not only for transportation but also for the pleasure of it. Some people want to drive along the seaside or on rural roads while enjoying their favorite landscape. We call such roadside landscapes "roadscapes." In such situations, it is not always the best solution to provide the shortest or the fastest route. An alternative solution is to provide routes with favored roadscapes even if they involve a detour.

Given starting and destination points, a route recommender system provides routes from the starting point to the destination point. The majority of traditional route recommender systems provide the shortest routes [3, 7], the fastest routes [4, 5, 9, 11, 12], or popular routes [1, 6, 8, 10]. As mentioned above, the shortest and the fastest routes do not always satisfy the user's demands. Systems that recommend popular routes provide routes many people are interested in. Wei et al. [8] extract popular routes by mining road links many people are interested in from their trajectories. Such route recommender systems consider the attractiveness of routes based on the wisdom of crowds, without considering the content features of routes.

In this paper, we focus on the roadscape as a route feature and propose the Roadscape-based Route Recommender System (R3), which provides diversified routes on the basis of roadscapes. Given starting and destination points, R3 provides four types of roadscape-based routes: rural-, mountainous-, waterside-, and urban-prior routes. For example, a user who likes waterside views can select waterside-prior routes from the four types of routes provided. To develop such a route recommender system, we have proposed a method for estimating roadscapes of given road links. In particular, we defined rural, mountainous, waterside, and urban elements as the roadscape elements, which are basic elements that compose a roadscape, through preliminary experiments. We defined a roadscape vector each of whose elements corresponds to a roadscape element and proposed a method for estimating such roadscape vectors for given road links. We presuppose that R3 is to be used on road network data with roadscape vectors.

Traditional route searching algorithms, such as the Dijkstra algorithm [2], are given the costs of road links and find a route that minimizes the sum of their costs. The simplest approach is to apply the traditional method and reduce the costs of the road links having the targeted roadscape elements. However, there is a high computational cost in applying such a method to a very large road network.

To reduce the computational cost, we propose a coarse-to-fine route search approach. We focus on the concept that similar roadscapes do not exist as fragments but in clusters. For example, there are some areas composed of similar roadscape elements, such as rural areas, mountainous areas, waterside areas, and urban areas. Based on this characteristic, we expect that we can reduce the computational cost by clustering similar roadscape areas in advance.

In this approach, we firstly extract areas—roadscape clusters—composed of similar roadscape elements by using a roadscape-based clustering method. Secondly, we create a roadscape cluster graph whose nodes correspond to the roadscape clusters and whose links correspond to the links between roadscape clusters. In the route searching process, given the roadscape cluster graph and starting and destination points, we roughly find four types of roadscape-based routes, which are the roadscape cluster sets passed through, one for each roadscape element; we call this the coarse-grained route search. Then, we find specific routes that connect the roadscape clusters in each type of route; we call this the fine-grained route search.

The contributions of this paper are as follows:

- We propose the Roadscape-based Route Recommender System (R3), which provides diversified roadscape-based routes, namely, rural-, mountainous-, waterside-, and urban-prior routes.
- To reduce the computational cost, we propose a coarse-to-fine route search approach that consists of a roadscape-based clustering method, a roadscape cluster graph, a coarse-grained route search, and a fine-grained route search.
- We evaluate the performance of R3 using network data for a real road. The results show that using coarse-grained route search can significantly reduce route search time.

2 PRELIMINARIES

Definition 1: Road network. A road network is a directed weighted graph $G = (V, E)$, where V is a set of road nodes and $E \subseteq V \times V$ is a set of road links. A road node $v_i \in V$ represents an intersection or an endpoint of a road. A road link $e_k = (v_i, v_j) \in E$ is a directed link from the starting node v_i to the ending node v_j. A road link e_k is assigned a cost w_k according to the length of the link.

Definition 2: Roadscape element. Roadscape elements are basic elements that compose a roadscape. We define four roadscape elements: rural, mountainous, waterside, and urban elements. These elements were selected by preliminary experimentation.[1]

Definition 3: Roadscape vector. A roadscape vector is defined as a four-dimensional probability vector each of whose elements corresponds to one of the respective roadscape elements. We define a roadscape vector of a road link e_i as $\boldsymbol{s}(e_i) = (s_i^r, s_i^m, s_i^w, s_i^u)$. Each element of the vector denotes the probability of how strongly e_i includes the corresponding roadscape element. Therefore, the sum of the values over all elements is 1.

Definition 4: Roadscape cluster. A roadscape cluster $C_j \in C$ is represented by a set of road links having similar roadscape vectors. A roadscape vector $\boldsymbol{s}(C_j)$ of roadscape cluster C_j is represented by the mean vector of the roadscape vectors of the road links included in cluster C_j. Therefore, we define $\boldsymbol{s}(C_j)$ as follows:

$$\boldsymbol{s}(C_j) = \frac{1}{|C_j|} \sum_{i \in C_j} \boldsymbol{s}(e_i). \tag{1}$$

Here, $|C_j|$ denotes the number of road links included in the roadscape cluster C_j.

Definition 5: Roadscape cluster graph. A roadscape cluster graph is a directed weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of roadscape clusters C_i and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of links between roadscape clusters. A link $l_k = (C_i, C_j) \in \mathcal{E}$ is a directed link from the starting node C_i to the ending node C_j. The road link l_k is assigned a cost vector $\boldsymbol{\omega_k} = (\omega_k^r, \omega_k^m, \omega_k^w, \omega_k^u)$ based on the roadscape vector $\boldsymbol{C_j}$ of ending roadscape cluster C_j. Each element of $\boldsymbol{\omega_k}$ denotes a cost for the corresponding roadscape; these are used for roadscape-based route searching. For example, ω_k^r is the cost referenced when searching for rural-prior routes.

Definition 6: Intra-cluster similarity of roadscape vector. The intra-cluster similarity is the mean similarity between all pairs of road links included in the cluster. We denote the intra-cluster similarity of roadscape cluster C_j as intra_sim(C_j). The value of intra_sim(C_j) is calculated as follows:

$$\text{intra\_sim}(C_j) = \frac{1}{n|C_j|} \sum_{i \in C_j} \sum_{k \in C_j} \cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_k)). \tag{2}$$

Here, e_i and e_k are road links included in cluster C_j, and n denotes the total number of links in the road network. The value of $\cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_k))$ is calculated as follows:

$$\cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_k)) = \frac{\boldsymbol{s}(e_i) \cdot \boldsymbol{s}(e_k)}{|\boldsymbol{s}(e_i)||\boldsymbol{s}(e_k)|}. \tag{3}$$

[1]The preliminary experimentation to select the roadscape elements was done via crowdsourcing. These four elements are specific to Japanese road network data. Details are outside the scope of this paper.

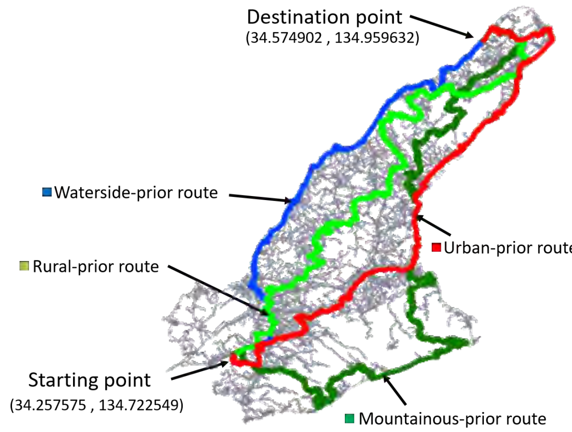


Figure 1: Recommended roadscape-based routes.

3 ROADSCAPE-BASED ROUTE RECOMMENDER SYSTEM

3.1 System Overview

Our proposed Roadscape-based Route Recommender System (R3) provides four types of roadscape-based routes: rural-, mountainous-, waterside-, and urban-prior routes. Figure 1 shows a result provided by R3. When a user inputs starting and destination points on the map, the four types of roadscape-based routes are provided in different colors.

It is assumed that R3 will be used with a road network with roadscape vectors. The steps of R3 are as follows:

(1) Generate roadscape cluster graph based on road network with roadscape vectors.
(2) Roughly find four types of roadscape-based routes in the roadscape cluster graph based on the starting and destination points that are input (coarse-grained route search).
(3) Find a detailed route that connects roadscape clusters in each type (fine-grained route search).
(4) Recommend four types of routes in different colors on the map.

Here, step (1) can be performed offline because this process does not depend on the inputs. In the next sections, we describe steps (1)–(3) in detail.

3.2 Generating Roadscape Cluster Graph

3.2.1 Roadscape-based Clustering. Given a road network, we form roadscape clusters based on proximities of pairs of road links and similarities between their roadscape vectors. Adjacent road links belong to the same cluster if their similarity is greater than or equal to a given threshold value. Figure 2 shows the result of applying roadscape-based clustering to the road network of Awaji

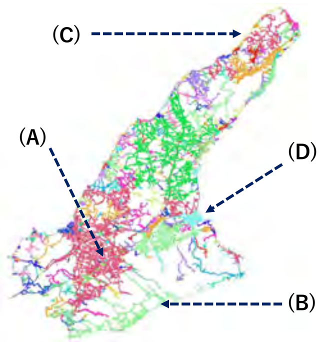


Figure 2: Result of applying roadscape-based clustering to the road network of Awaji Island, Japan. Each color corresponds to a given cluster.

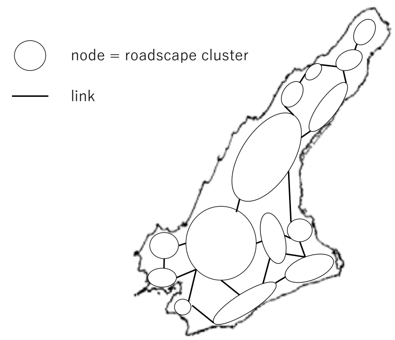


Figure 3: Example of a roadscape cluster graph created for Awaji Island's road network.

Island, Japan. Here, area A corresponds to a rural area, area B corresponds to a mountainous area, area C corresponds to a waterside area, and area D corresponds to an urban area.

Algorithm 1 shows the pseudocode for roadscape-based clustering. We explain the clustering process as performed by Algorithm 1 as follows:

Algorithm 1 Roadscape-based clustering.

```
Require: Target link e_i, Cluster ID k
function ROADSCAPECLUSTERING(e_i, k)
    Cluster ID of e_i ⇐ k
    linkList ⇐ getLink(e_i): Get links adjacent to e_i.
    for each e_j in linkList
        if Cluster ID of e_j = 0 then
            if cos(s(e_i), s(e_j)) >= α then
                roadscapeClustering(e_j, k)
            end if
        end if
    end for
    return 0
end function
```

We randomly select a road link from the road network. Let e_i be the target link, and let e_j be one of the links adjacent to e_i. Here, if two links are connected to a common node, the links are considered adjacent. Furthermore, let $\boldsymbol{s}(e_i)$ and $\boldsymbol{s}(e_j)$ be roadscape vectors of the respective links.

The roadscape-based clustering algorithm is called as `roadscapeClustering`(e_i, k). First, add k as the cluster ID of e_i. Second, get all links adjacent to e_i, and set them into `linkList`. For each link $e_j \in$ `linkList`, perform the following process. If a cluster ID has not been assigned to e_j, $\cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_j))$ (Equation (3)) is calculated. If $\cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_j))$ is greater than or equal to the threshold α, cluster ID k of e_i is added as the cluster ID of e_j. Furthermore, `roadscapeClustering`(e_j, k) is recursively called. The above process is repeated until the cluster ID has been added to all of the links in the road network.

We define the roadscape cluster obtained by the above process as $C_k \in C$, where k corresponds to the cluster ID. In addition, roadscape vector $\boldsymbol{s}(C_k)$ of cluster C_k is calculated by Equation (1).

3.2.2 Generating Roadscape Cluster Graph. After extracting the roadscape clusters, we create the adjacency matrix for all roadscape clusters. The adjacency matrix for the roadscape clusters is represented as the $|C| \times |C|$ matrix $\mathcal{A} = [a_{ij}]_{|C|\times|C|}$. If $a_{ij} = 1$, clusters C_i and C_j have at least one common node; otherwise, they do not have a common node.

We then create the roadscape cluster graph based on the adjacency matrix. Figure 3 gives an example of the roadscape cluster graph created for Awaji Island's road network. Here, a node in the roadscape cluster graph corresponds to a roadscape cluster, and a link corresponds to the adjacency relationship between clusters.

3.2.3 Assigning Costs to Roadscape Cluster Graph. In order to execute the coarse-grained route search described in the next section, we assign costs to the links of the roadscape cluster graph in advance. A link cost is calculated based on the roadscape vector of the roadscape cluster corresponding to the link's destination. If the targeted roadscape element of the next roadscape cluster destination is emphasized, let its link cost be lower; on the other hand, if it is not emphasized, let its link cost be higher. For example, for a case in which a rural element is targeted, if the rural element of the next roadscape cluster destination is emphasized, let its link cost be lower; otherwise, let its link cost be higher. By assigning costs in such a way, the route to the roadscape cluster where the rural element is emphasized is more likely to be chosen in the route search.

A cost vector $\boldsymbol{\omega}_k$ of link $l_k = (C_i, C_j)$ is calculated as follows:

$$\boldsymbol{\omega}_k = d_k(1 - \boldsymbol{s}(C_j)^2). \tag{4}$$

Here, d_k is the length of link l_k.

3.3 Coarse-grained Route Search

As the first search, we execute the coarse-grained route search method. This method roughly finds four types of roadscape-based routes in the roadscape cluster graph. The process is as follows:

(1) Given starting and destination points, get roadscape clusters and starting and destination clusters, which include the starting and destination points, respectively.

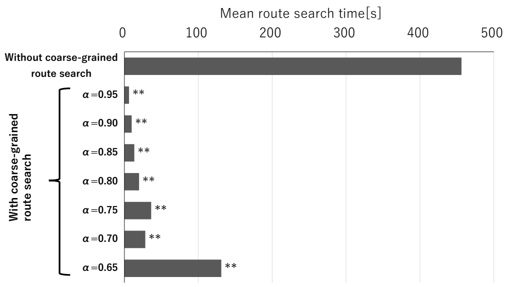


Figure 4: Comparison of route search times.

(2) For the targeted roadscape element, find a route that minimizes the sum of the link costs related to the targeted elements using Dijkstra's algorithm [2] on the roadscape cluster graph.
(3) Repeat step (2) for each roadscape element.

Thus, we obtain four types of coarse-grained routes as the roadscape cluster sets that are passed through for each roadscape element.

3.4 Fine-grained Route Search

As the second search, we execute the fine-grained route search method for each coarse-grained route. This method finds detailed routes that connect roadscape clusters. The process for each targeted element is as follows:

(1) Find common road nodes of each adjacent cluster in the roadscape cluster sets captured by the coarse-grained route search.
(2) Find the shortest route from the starting point to the first common road node that is adjacent to the next cluster.
(3) While there are common road nodes, find the shortest route from the common road node to the next common node.
(4) Find the shortest route from the last common node to the destination point.
(5) Generate a route that connects all the routes obtained.

Here, we again use Dijkstra's algorithm [2] to find the shortest routes. Finally, we obtain four types of fine-grained routes.

4 RESULTS

In this section, we evaluate the performance of the proposed R3 method using network data for a real road in Awaji Island, Japan. The road network data are derived from OpenStreetMap,[2] and they include 102,506 road nodes and 212,050 road links for the area of Awaji Island. For this area, roadscape vectors for all road links are available on the web.[3]

R3 introduces a coarse-grained route search as preprocessing to reduce the route search time instead of performing a route search on all road links. In this section, we compare the route search times using coarse-grained route search with those not using it.

First, we prepare the following five pairs of starting and destination points.

(a) $(34.257575, 134.722549) \rightarrow (34.574902, 134.959632)$
(b) $(34.317774, 134.676412) \rightarrow (34.348304, 134.896255)$
(c) $(34.499798, 134.938260) \rightarrow (34.293801, 134.788816)$
(d) $(34.545838, 134.923368) \rightarrow (34.440009, 134.912038)$
(e) $(34.208185, 134.814500) \rightarrow (34.430861, 134.830634)$

For each pair, we execute the route search algorithm that emphasizes each roadscape element and measure the route search time. We regard this execution as one trial. We execute this trial ten times for each pair and calculate the mean of the route search times across trials.

We implemented the route search algorithm using Java and managed the road network data using PostgreSQL 9.5. We conducted experiments on a computer equipped with an Intel Core i5-6200U CPU (2.8 GHz), 8 GB memory, 256 GB SSD, and Linux Mint 18.2.

Figure 4 shows the mean route search times for methods with and without coarse-grained route search. For the method with coarse-grained route search, the figure includes the route search time for each value of α. $**$ indicates that a significant difference ($p < 0.01$) could be confirmed when comparing with the method without coarse-grained route search by the paired t-test (one-sided test). We can see from Figure 4 that the route search time can be shortened by using coarse-grained route search. The figure also shows that the higher the value of α was, the shorter the route search time was. In particular, when $\alpha = 0.95$, the search time with coarse-grained route search was 6.24 s, whereas it was 456 s when coarse-grained route search was not used. Consequently, we can say that the use of coarse-grained route search can significantly reduce route search time.

5 CONCLUSIONS

In this paper, we have proposed a Roadscape-based Route Recommender System (R3) that provides diversified roadscape-based routes. Given starting and destination points, R3 provides four types of roadscape-based routes: rural-, mountainous-, waterside-, and urban-prior routes. To reduce computational costs, we proposed a coarse-to-fine route search approach that consists of a roadscape-based clustering method, a roadscape cluster graph, a coarse-grained route search, and a fine-grained route search.

We evaluated the performance of R3 using real road network data with roadscape vectors in the area of Awaji Island. The results show that using coarse-grained route search can significantly reduce route search time. In the future, we will conduct user tests to evaluate our system from the users' perspective.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP15K12151 and JP16HO593.

REFERENCES

[1] Zaiben Chen, Heng Tao Shen, and Xiaofang Zhou. 2011. Discovering Popular Routes from Trajectories. In *Proceedings of the 2011 IEEE 27th International Conference on Data*, Vol. 4. 900–911.
[2] Edsger Wybe Dijkstra. 1959. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* 1 (1959), 269–271.
[3] Andrew V Goldberg and Chris Harrelson. 2005. Computing the shortest path: A search meets graph theory. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. 156–165.

[2] https://www.openstreetmap.org/
[3] https://zenodo.org/record/1405255#.W4Yyb-j7T-g

[4] Hector Gonzalez, Jiawei Han, and Xiaolei Li. 2007. Adaptive fastest path computation on a road network: a traffic mining approach. In *Proceedings of the 33rd international conference on Very large data bases*. 794–805.

[5] Evangelos Kanoulas, Yang Du, Tian Xia, and Donghui Zhang. 2006. Finding fastest paths on a road network with speed patterns. In *Proceedings of the 22nd International Conference on Data Engineering*, Vol. 2006. 10.

[6] Wuman Luo, Haoyu Tan, Lei Chen, and Lionel M. Ni. 2013. Finding time period-based most frequent path in big trajectory data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 713–724.

[7] Michalis Potamias, Francesco Bonchi, Carlos Castillo, and Aristides Gionis. 2009. Fast shortest path distance estimation in large networks. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 867–876.

[8] Ling-yin Wei, Wen-chih Peng, Bo-chong Chen, and Ting-wei Lin. 2010. Eleventh International Conference on Mobile Data Management PATS : A Framework of Pattern-Aware Trajectory Search. In *Mobile Data Management*. 362–377.

[9] Ling-yin Wei, Wen-chih Peng, Chun-shuo Lin, and Chen-hen Jung. 2009. Exploring Spatio-Temporal Features. In *Advances in Spatial and Temporal Databases, Lecture Notes in Computer Science*. 399–404.

[10] Ling-Yin Wei, Yu Zheng, and Wen-Chih Peng. 2012. Constructing popular routes from uncertain trajectories. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 195–203.

[11] Jing Yuan, Yu Zheng, Chengyang Zhang, and Wenlei Xie. 2010. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 99–108.

[12] Jing Yuan, Yu Zheng, Liuhang Zhang, XIng Xie, and Guangzhong Sun. 2011. Where to find my next passenger. In *Proceedings of the 13th international conference on Ubiquitous computing*. 109–118.

Understanding Customer Choices to Improve Recommendations in the Air Travel Industry

Alejandro Mottini
Amadeus SAS
Sophia Antipolis, France

Alix Lhéritier
Amadeus SAS
Sophia Antipolis, France

Rodrigo Acuna-Agost
Amadeus SAS
Sophia Antipolis, France

Maria A. Zuluaga
Amadeus SAS
Sophia Antipolis, France

ABSTRACT

Recommender systems aim at suggesting relevant items to users to support them in various decision-making processes, on the basis of available information on items or users. In the latter, the customer's interests and tastes can be learnt and expressed using historical browsing data, purchase histories, and even other non-traditional data sources such as social networks. Despite its proven success in the on-line retailing industry, in electronic commerce and, even tourism, recommender systems have been less popular in flight itinerary selection processes. This could be partially explained by the fact that customers' interests are only expressed as a flight search request. As a result, this problem has been historically tackled using classical Discrete Choice Modelling techniques and, more recently, through the use of data-driven approaches such as Machine and Deep Learning techniques. At Amadeus, we are interested in the use of choice models with recommender systems for the problem of airline itinerary selection. This work presents a benchmark on three family of methods to identify which is the most suitable for the problem we tackle.

KEYWORDS

Choice Modeling; Choice-based Recommendations; Air Travel Industry

1 INTRODUCTION

In the recent years, recommender systems (RecSys) have proven invaluable for solving problems in the on-line retail industry and e-commerce[15]. While tourism has not been the exception to this success [3], with applications covering almost every area of the travel and hospitality industry [14], RecSys have been less popular on the airline itinerary decision-making process. This can be explained by two factors. On one hand, the available information about users and items is not as rich as for most RecSys in tourism. In the traveller's flight itinerary choice problem, *i.e.* the task of selecting a flight given a proposed list of itinerary recommendations, the user's interests are only expressed as a flight search request, user sessions are usually anonymous and there is no user history in the travel provider's databases. Therefore, classical RecSys algorithms cannot be applied directly.

On the other hand, RecSys techniques suffer from a lack of theoretical understanding of the underlying behavioural process that led to a particular choice [6] by seeing the decision-making process as a *black box* [7]. Collaborative and content-based methods recommend items based on similarities among users or items but, cannot provide further insight. In the flight industry, it is key to understanding passenger behaviour and their flight itinerary preferences. Players in the sector use this knowledge to adapt their offers to market conditions and customer needs, thus having an impact on airline's revenue management and price optimisation systems [4].

To tackle the flight itinerary choice problem and overcome these limitations, the airline industry has historically resorted to Discrete Choice Modeling (CM). Due to its good performance, efficiency and ease of interpretation, the Multinomial Logit model (MNL) [11], a specific CM technique is the most popular approach for the flight itinerary choice problem. In spite of its numerous advantages, CM also presents some weaknesses. For instance, MNL only considers linear combinations of the input features, limiting its predictive capability and requiring expert knowledge to perform feature engineering. Also, they lack the flexibility to handle collinear attributes and correlations between options and it is difficult to model individual's heterogeneities. These shortcomings might be overly restrictive or affect performance [12]. As an example, industrial applications require to develop different models for distinct markets. In the case of the flight itinerary choice prediction problem, this involves estimating models at a city-pair level [5] and/or customer demographic segments [19].

In an effort to cope with CM limitations, recently machine learning and deep learning techniques have been proposed. These algorithms can more easily model non-linear relationships and handle correlated features, and have more modelling power which allows to predict choices on an individual level, thus improving the prediction performance.

Inspired by the work from Chaptini [6], at Amadeus we are working towards the use of CM with recommender systems for the problem of airline itinerary selection. Combining the two approaches should leverage the strengths of both, leading to robust and scalable, but more interpretable models. In this first work, we seek to explore, evaluate and compare three different CM models which can be used as the predictive back-bone of a choice-based RecSys framework. In the remainder of this paper, first we present the theoretical background of CM and demonstrate why CM can be seen as a RecSys problem. Then, we present our experimental setup by describing the data, the evaluated algorithms and the performance measures.

2 BACKGROUND

In this section, first we provide a brief background on classical discrete choice modelling theory and then show how it is equivalent to the recommendation problem.

2.1 Discrete Choice Models.

CM defines four basic components: 1) the decision-maker, 2) the alternatives, 3) the attributes, and 4) the decision rules [2]. Formally stated, a decision-maker $i \in I$ chooses from a choice set A_i composed of J_i alternatives, with with $j \in \{1, \ldots, J_i\}$ the index of the j^{th} alternative. For the sake of simplicity and without loss of generality, we will refer to the number of alternatives simply as J, although decision-makers might not be faced with the same set and/or number of alternatives. The decision-maker i obtains an utility U_{ij} from each j and chooses alternative $\hat{j}$ if and only if:

$$U_{i,\hat{j}} \geq U_{ij};\ \forall\, j \in A_i. \tag{1}$$

The utility function is unknown and not observable. However, as it is possible to determine the attributes $\mathbf{x}_{ij}$ perceived by decision-maker i for each j, as well as S_i the vector of characteristics of i, there exists a function $V(\cdot)$ which relates the observed features to the decision-maker's utility:

$$V_{ij} = V(\mathbf{X}_{ij}), \tag{2}$$

where V_{ij} is referred to as the representative utility and $\mathbf{X}_{ij} = h(\mathbf{x}_{ij}, \mathrm{S}_i)$, a simplified representation of $\mathbf{x}_{ij}$ and S_i through the use of any appropriate vector valued function h. V_{ij} is generally a linear combination of the features. For example, if an airline is trying to predict which itinerary a user will choose, a very simple model could be:

$$V_{ij} = a * price_{ij} + b * tripDuration_{ij}$$

with a, b parameters of the model to be estimated, and which are commonly refered to as β.

Since there are aspects of the utility function that cannot be observed, $V_{ij} \neq U_{ij}$. To reflect uncertainty, the utility can be modelled as a random variable,

$$U_{ij} = V_{ij} + \varepsilon_{ij}, \tag{3}$$

where ε_{ij} is a random variable that captures the unknown factors that affect U_{ij}. As U_{ij} is now a random variable, the decision rule needs to be expressed as the probability that decision-maker i chooses the k^{th} alternative:

$$P(k|A_i) = P(U_{ik} \geq U_{ij};\ \ \forall j \in A_i). \tag{4}$$

By replacing U_{ij} accordingly:

$$P(k|A_i) = P(V_{ik} - V_{ij} \geq \varepsilon_{ij} - \varepsilon_{ik};\ \ \forall j \in A_i). \tag{5}$$

Different assumptions about the random term ε_{ij} and the deterministic term V_{ij} produce specific models.

2.2 Choice-based Recommender Systems.

Given a set A_i of J available items presented to a user i, the recommender problem can be seen as an optimisation task that first estimates the utility of each item $j \in A_i$, and then chooses the item $\hat{j}$ that maximizes an utility function $U(i, j)$, representing the user's utility on any item j [1]:

$$\hat{j} = \underset{j \in A_i}{\arg\max}\ \ U(i, j). \tag{6}$$

Conceptually this is the same optimisation problem as that one formulated by choice theory [2], and described previously in this section. Equation (6) is equivalent to choosing the alternative with the highest utility for a decision-maker, in choice modelling theory. More formally:

$$\hat{j} = \underset{j \in A_i}{\arg\max}\ \ U(i, j) \Leftrightarrow U_{i\hat{j}} \geq U_{ij};\ \forall\, j \in A_i, \tag{7}$$

which implies that the recommendation problem can be seen as a choice prediction problem. Therefore, the models and techniques developed in CM can be applied to RecSys.

3 MATERIALS AND METHODS

3.1 Data

Experiments were conducted on real datasets of flight search logs and bookings from MIDT, an Amadeus database containing bookings from over 93000 travel agencies.

Bookings are stored using Personal Name Records (PNR), which are created at reservation time by airlines or other air travel providers, and are then stored in the airline's or Global Distribution System (GDS) data centers. PNRs contain the travel itinerary of the passenger, personal and payment information, and/or additional ancillary services sold with the ticket. As these only contain information about the purchased ticket (final choice), and not about the alternatives considered before the purchase, we must also consider flight search logs. These contain both itinerary requests (origin, destination and dates), and the different alternatives presented to the passenger.

Both data sources are combined into a final dataset containing the alternatives presented to each user and their final choice (Figure 1). The matching process is in itself a challenging problem due to the high volume of data (i.e., around 100 GB of daily search logs) and to the difference in data sources and formats. Moreover, the process cannot be perfectly accurate since there is not a direct link between the two data sources and booking/search times differ. An approximate matching is performed using data fields which are shared between booking and logs (i.e. origin, destination, time and booking agency).

The choice set presented to a user, which we denote a session, contains up to 50 itineraries. The features used for each alternative are summarized in Table 1. The considered dataset contains 33951 sessions split into training/tests sets.

3.2 Algorithms

Methods from three different families of algorithms, classical CM, machine learning- and deep learning-based CM, are explored.

3.2.1 Classical CM. Two classical CM approaches are considered: The Multinomial Logit (MNL) model [11], perhaps the most common CM model, and Latent class choice models (LCM) [8]. McFadden [11] demonstrated that if ε_{ij} is an i.i.d. Gumbel random variable, the probability that a decision-maker i chooses the

Table 1: Feature set classified according to owner (individual or alternative) and data type (numerical, categorical, binary or time).

Owner	Data Type	Features
Individual	Categorical	Origin, destination, office
	Numerical	Days to trip, Trip weekday
	Binary	Stays Saturday, Continental trip, Domestic trip
Alternatives	Categorical	Airline of first flight
	Numerical	Price, Stay length, Trip duration, Connections, Num. of airlines
	DateTime	Arrival time, Departure time

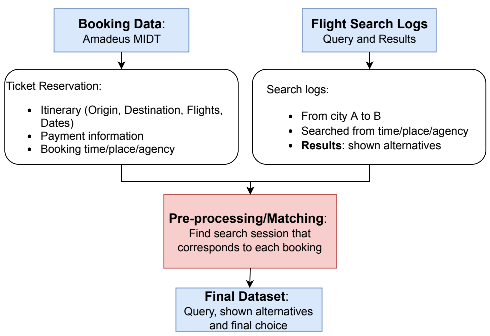


Figure 1: Dataset generation through MIDT bookings and search log matching.

alternative k (Eq. 5), the logit choice probability, is given by:

$$P(k|A_i) = \frac{\exp(V_{ik})}{\sum_{j \in J} \exp(V_{ij})}. \quad (8)$$

LCMs have been proposed to capture unobserved heterogeneity. Under LCM, the probability of choosing an alternative k can be expressed as:

$$P(k|\mathbf{X}_{ij}) = \sum_{q=1}^{Q} P(k|\mathbf{X}_{ij}; \beta_q, A_q) P(q|\mathbf{X}_{ij}; \theta) \quad (9)$$

where Q is the number of latent classes, β_q are the choice model parameters specific to class q, A_q is the choice set specific to class q, θ is an unknown parameter vector, and $\mathbf{X}_{ij}$ the simplified vector representation of attributes of alternatives and characteristics of decision-maker i.

Finally, both MNL and LCM models are optimized using maximum likelihood estimation as they can not be solved in a closed form.

3.2.2 ML. Lheritier *et al.* a have proposed machine-learning based CM (ML) [10] technique which formulates the choice modelling problem as a supervised learning one through the use of Random Forests (RF), a learning algorithm based on an ensemble of decision trees. The training data consists of the set of sample pairs $\mathcal{T} = \{(\mathbf{X}_{ij}, y_{ij})\}$ [1], with y_{ij} the binary indicator of whether decision maker i chooses the j-th alternative. As RF assumes independence of the samples, at training stage, every $\mathbf{X}_{ij}$ is assumed i.i.d., even if they belong to the same decision-maker. At prediction, each unseen alternative $\mathbf{X}_{ij}$ is propagated through the trained forest to obtain the posterior probability of being chosen:

$$P(y_{ij}|\mathbf{X}_{ij}) = \frac{1}{T} \sum_{t=1}^{T} P_{l_t}(y_{ij}(\mathbf{X}_{ij}) = 1) \quad (10)$$

where T denotes the number of trees and $P_{l_t}(\cdot)$ denotes the posterior probability function of a leaf node l in tree t. However, the alternatives associated to an individual's session cannot be treated as independent. There is an inherent dependence among them: only one alternative per session can be selected. To cope with this, the predicted probabilities are considered scores used to rank the alternatives. More formally, the index $\hat{j}$ of the selected alternative $a_{\hat{j}}$ by decision-maker i is:

$$\hat{j} = \underset{1 \leq j \leq J}{\arg\max}\; P(y_{ij}|\mathbf{X}_{ij})$$

3.2.3 DL. The assessed Deep learning choice modeling (DL) method [13] is based on an encoder-decoder network architecture using a modified pointer-network mechanism [18]. As with ML, the model is trained to predict the chosen alternative using a supervised learning approach. However, DL does not break the i.i.d. assumption among samples, as ML-based CM does. Given the sequential nature of pointer networks, sessions are represented as sequences of itineraries, $Z = \{\mathbf{X}_{i1}, ..., \mathbf{X}_{iJ}\}$, which are fed sequentially to the model. The encoder network "encodes" the input into a hidden (encoder) state e. The decoder network will use the encoded information to output a vector u. Finally, a softmax function use the decoder's output to estimate the posterior probability of being chosen for the k^{th} element in the input sequence Z :

$$P(y_k = 1|Z) = \frac{\exp(u_k)}{\sum_{j=1}^{J} \exp(u_j)} \quad (11)$$

with $u_k = d^T W_1 e_k$, the pointer vector to the k^{th} element of Z, e_k the k^{th} encoder state, $d = \tanh(W_2 e_J)$ the decoder, W_1, W_2 learnable parameters and y_k the binary indicator of whether k was chosen ($y_k = 1$) or not. $P(y_k = 1|Z)$ can be interpreted as an estimate of $P(k|A_i)$.

3.3 Performance measurement

We used Top-N accuracy to asses and compare the models. Top-N accuracy evaluates if the user's choice is among the top-N predicted alternatives. It is equivalent to the commonly used top-N error in image classification [16], as it can be formulated in terms of the latter as:

$$accuracy = 1 - error$$

4 RESULTS

Figure 2 presents the Top-N accuracy for MNL, ML and DL methods. Overall, DL presents the highest accuracies across all values of N. These results are confirmed, in more detail, in Table 2 where Top-1, 5 and 15 accuracies are detailed. Top-15 accuracy has a particular importance for ranking flight search recommendations since most websites show approximately 15 results per page.

[1]In the context of RF, $\mathbf{X}_{ij}$ referred to as the feature vector of a sample

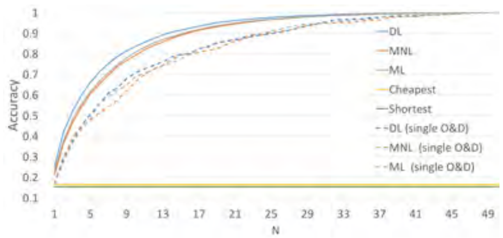


Figure 2: Top-N accuracy using the full data set (solid line) and a subset of the dataset consisting of a single origin/destination (O&D) pair (dashed line).

Table 2: Top-N with $N = 1, 5$ and 15. The best result for each N is presented in bold. Trivial choices of cheapest and shortest flight are included for reference.

Method	Top-1.	Top-5	Top-15
DL	**25.3**	**66.37**	**93.1**
ML	23.1	61.7	92.9
MNL	21.2	60.6	86.4
Cheapest	16.4	16.4	-
Shortest	15.4	15.4	-

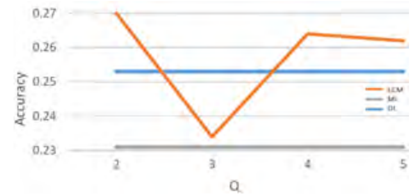


Figure 3: Top-1 accuracy of LCM, , as a function of the number of latent classes Q, compared to ML- and DL-based approaches.

Figure 4: Top 8 feature importance for the ML method.

To simulate data segmentation, a second experiment was performed in a simplified subset containing a single origin-destination (O&D) pair chosen at random. This resulted in 1617 decision-makers (users) with an associated booking to the O&D. The Top-N accuracy curve (Figure 2 dashed lines) shows how the difference in performance between the methods is less significant w.r.t. that one using the full data set. Despite MNL being the simplest method, results show that, on simpler datasets, it is able to perform as well as more complex methods.

This behaviour explains the motivation behind dataset pre-segmentation often used in classical CM. This is further confirmed by investigating the performance of LCM, as a function of the number of latent classes Q. Figure 3 reports top-1 accuracy of LCM, ML and DL, and demonstrates how it is possible to increase classic CM accuracy in complex data through a good estimation of Q. While MNL reported accuracies lower than ML and DM, LCM can outperform them when Q is estimated correctly. This improvements comes, however, at some cost: LCM requires additional hand engineered features to achieve the segmentation and a good choice of Q.

Although ML and MNL are not as accurate as DL, they have the advantage of having less hyper-parameters to tune. Moreover, they are more interpretable than DL. ML methods based on RF are known for their capacity to provide information on feature importance (Figure 4). This type of information can help to understand the rationale behind the decision-maker's choices, which can be important for some applications in the air travel industry.

5 FINAL REMARKS

RecSys research has so far predominantly focused on optimizing the algorithms used for generating recommendations to increase precision [9]. Precision measures how well the suggested alternatives match a decision-maker's profile based on previous data. While this is an important criterion, its limited assessment of a recommender quality has been criticized for not taking the decision-makers' situational needs into account [9]. Due to their well-known readability, Discrete Choice Modelling appears as a natural alternative to overcome this current limitation of RecSys. However, despite CM being a well-studied problem in various fields of research, literature on its use with recommender systems is very scarce. Existing works have adopted classical CM in combination RecSys [6, 17], while suggesting CM as a promising paradigm in the field of RecSys.

However, classical choice models tend to suffer from scalability issues as expert knowledge is usually required for model optimisation. ML- and DL-based [10, 13] choice models are non-parametric approaches that overcome this limitation, easing the deployment of choice-based RecSys at large scale. On the down side, model readability can diminish. Although this might not be relevant for some applications, understanding the reasons behind a decision-maker's choice is of high relevance in the air travel industry. ML-based methods appear to be a suitable compromise into readability but, they make strong assumptions on the independence of data that is arguable. Overall, it is possible to say that there is no ideal method and that the selection of one might depend on the specific recommendation application that they target. As a guideline, Table 3 summarises the strengths and pitfalls of the different methods here evaluated when considering choice-based RecSys.

At Amadeus, we work towards the development of informative, readable and interpretable RecSys that suit the needs of the air travel industry. Our hypothesis is that the combination of discrete choice modeling with RecSys can provide improvements to current systems in the air travel industry by keeping readability while improving performance. In that sense, an ML method like the random

Table 3: Advantages and disadvantages of the different families of CM methods.

Method	Advantages	Disadvantages
CM	• Simple and interpretable • Accurate on simple cases	• Feature engineering is required • Limited in handling big data
ML	• Interpretable • Accurate • Suitable for big data • Handles non-linear and latent relationships	• Assumes independence of samples • Feature engineering might be required
DL	• No assumptions on data • Highly accurate • Suitable for big data • Handles non-linear and latent relationships	• Non-interpretable • Many hyper-parameters • Computationally expensive

forests evaluated here represents a good compromise and a promising path to pursue in what we are looking for. On one hand, the method provides information on the relevance of features. On the other one it avoids the limitations of classical CM models. In that sense, although DL approaches have higher accuracy, they are not as advantageous given their limited interpretability.

This work represents an initial benchmark that evaluates three families of CM methods in the context of flight itinerary selection/recommmendation. Our future work will focus in the development of a unified framework that can leverage the strengths of the explored CM methods.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749.

[2] Moshe Ben-Akiva and Michel Bierlaire. 1985. *Discrete choice analysis: theory and application to travel demand.* MIT press, Cambridge, MA.

[3] Joan Borràs, Antonio Moreno, and Aida Valls. 2014. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications* 41, 16 (2014), 7370–7389.

[4] J. Broder and P. Rusmevichientong. 2012. Dynamic pricing under a general parametric choice model. *Operations Research* 60, 4 (2012), 965–980.

[5] Judit G Busquets, Antony D Evans, and Eduardo Alonso. 2016. Predicting Aggregate Air Itinerary Shares Using Discrete Choice Modeling. In *16th AIAA Aviation Technology, Integration, and Operations Conference.* 4076.

[6] Bassam H. Chaptini. 2005. *Use of discrete choice models with recommender systems.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[7] Li Chen, Marco de Gemmis, Alexander Felfernig, Pasquale Lops, Francesco Ricci, and Giovanni Semeraro. 2013. Human Decision Making and Recommender Systems. *ACM Transactions on Interactive Intelligent Systems* 3, 3 (2013), 1–7.

[8] William H Greene and David A Hensher. 2003. A latent class model for discrete choice analysis: contrasts with mixed logit. *Transportation Research Part B: Methodological* 37, 8 (2003), 681–698.

[9] Joseph A. Konstan and John Riedl. 2012. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 101–123.

[10] Alix Lhéritier, Michael Bocamazo, Thierry Delahaye, and Rodrigo Acuna-Agost. 2018. Airline Itinerary Choice Modeling using Machine Learning. *International Journal of Choice Modeling* (2018), https://doi.org/10.1016/j.jocm.2018.02.002.

[11] Daniel McFadden. 1973. Conditional Logit Analysis of Qualitative Choice Behaviour. In *Frontiers in Econometrics*, P. Zarembka (Ed.). Academic Press New York, New York, NY, USA, 105–142.

[12] D. McFadden. 2001. Economic choices. *The American Economic Review* 91, 3 (2001), 351–378.

[13] Alejandro Mottini and Rodrigo Acuna-Agost. 2017. Deep Choice Model Using Pointer Networks for Airline Itinerary Predictions. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, New York, NY, USA, 1575–1583.

[14] Julia Neidhardt, Tsvi Kuflik, and Wolfgang WÃűrndl. 2018. Special section on recommender systems in tourism. *Information Technology & Tourism* 19, 1-4 (2018), 83–85.

[15] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). 2011. *Recommender Systems Handbook.* Springer Nature, New York.

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[17] Paula Saavedra, Pablo Barreiro, Roi Duran, Rosa Crujeiras, María Loureiro, and Eduardo Sánchez Vila. 2016. Choice-Based Recommender Systems. In *RecTour@ RecSys - Workshop on Recommenders in Tourism held in conjunction with the 10th ACM Conference on Recommender Systems (RecSys).* CEUR Workshop Proceedings, Boston, MA, USA, 38–46.

[18] O. Vinyals, M. Fortunato, and N. Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems (NIPS 2015).* Curran Associates, Inc., Montreal, Canada, 2692–2700.

[19] V. Warburg, C. Bhat, and T. Adler. 2006. Modeling demographic and unobserved heterogeneity in air passengersâĂŹ sensitivity to service attributes in itinerary choice. *Journal of the Transportation Research Board* 1951 (2006).

TourExplain: A Crowdsourcing Pipeline for Generating Explanations for Groups of Tourists

Öykü Kapcak, Simone Spagnoli, Vincent Robbemond, Soumitri Vadali, Shabnam Najafian, Nava Tintarev
Delft University of Technology, Delft, the Netherlands
{kapcak,s.spagnoli,v.t.robbemond,s.vadali}@student.tudelft.nl
{s.najafian,n.tintarev}@tudelft.nl

ABSTRACT

When a group is traveling together it is challenging to recommend an itinerary consisting of several points of interest (POIs). The preferences of individual group members often diverge, but it is important to keep everyone in the group satisfied during the entire trip. We propose a method to consider the preferences of all the people in the group. Building on this method, we design explanations for groups of people, to help them reach a consensus for places to visit. However, one open question is how to best formulate explanations for such sequences. In this paper, we introduce *TourExplain*, an automated crowdsourcing pipeline to generate and evaluate explanations for groups with the aim of improving our initial proposed explanations by relying on the wisdom of crowds.

CCS CONCEPTS

• Information systems → Crowdsourcing; **Recommender systems**; **• Human-centered computing → Natural language interfaces**; **Empirical studies in HCI**;

KEYWORDS

Explanations; Crowdsourcing; Crowdworking; Group recommendation; Tourism; Sequences

1 INTRODUCTION

Recommender systems are decision support systems which help users to find one or more items in a large space of possible options that best fit their wishes and needs. The main focus of current recommender systems is to propose items to individual users. However, in tourism people often consume several items, and often do so in groups rather than individually.

A group traveling together can be recommended an itinerary consisting of several points of interest (POIs). However, reaching a consensus is difficult, and often compromises need to be made. Such compromises can potentially help users expand their tastes. Mary's preferred POI may become John's new favorite spot! Compromises can also lead to rejection of the recommended items. One way to avoid this is to explain recommendations that are surprising; or even expected to be disliked; by an individual user [12]. In addition, there are many ways to formulate explanations for groups, but few guidelines for generating such explanations. To address these challenges, we present a novel crowdsourcing pipeline for generating and evaluating group explanations.

* The first to fourth authors contributed equally to this work.

2 RELATED WORK

This work builds on two strands of research, namely 1) explanations for group recommendations and 2) crowdsourcing for improving the explanation text.

2.1 Explanations

A group traveling together can be recommended an itinerary consisting of several points of interest (POIs). To keep the group satisfied during the entire sequence of recommendations (e.g., POIs), we need to consider the preferences of all the people in the group [5]. This can be challenging when the preferences of individual group members diverge. An explanation in such contexts can assist users reach a consensus for places to visit.

Ardissono et al. [1] developed a handheld recommender system for sightseeing destinations and itineraries for heterogeneous tourist groups. This system supplied explanations based on the properties of items but did not consider the need to support consensus. Moreover, Nguyen and Ricci also combined user preferences generated by the interactions between group members. Although they studied group decision making and consensus, they have not studied explanations [13].

Masthoff et al. [10] suggest several *preference aggregation strategies*. These have as input a set of predictions for all users in a group for a set of items, and have as output a sequence of recommended items. In our previous work, we built on this work and designed explanations for groups of people that helped them reach a consensus [12]. One open question is how to best formulate explanations for such sequences. In this work, we therefore aim to improve our initial proposed explanations by relying on human wisdom using crowdsourcing.

2.2 Crowdsourcing

Crowdsourcing is a practice for solving computationally hard tasks by assigning them to an undefined (and generally large) network of people in the form of an open call, usually through online platforms (Mechanical Turk[1], FigureEight[2], etc.). This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals (crowdworkers) [7]. Crowdsourcing approaches are used for creating content or generating ideas with the contribution of a crowd. The approach proposed in this paper is to use the wisdom of crowds to generate and improve explanation text for end-users. This idea is similar to previous work

[1]https://www.mturk.com/, retrieved July 2018
[2]https://www.figure-eight.com/, retrieved July 2018

which used crowdsourcing to find better formulations for numerical expressions [2]. This previous work used templates to collect simple sentences (perspectives) from workers to make numerical expressions easier to understand. Finally, they evaluated the effectiveness of these perspectives on everyday readers' numerical comprehension.

Similarly, other authors proposed a model to generate personalized natural language explanations in the movie domain [4]. The crowdworkers were provided by quotes extracted from online movie reviews and the user rating history. Compared to our work, these explanations were designed for the movie domain and for individual users rather than group recommendations. Another difference in the design pattern: we specify specific criteria (based on Gricean Maxims [6]) in our all three steps: Find, Fix and Verify steps. In the finding step to give crowdworkers clear guidelines for finding any shortcomings in terms of these criteria; in the fixing step to give them clear guidelines for improving the explanations; in the verification step for validating the explanation.

Bernstein et al. [3] also applied crowd-sourced contributions to help humans write and edit their work. *Soylent* is a language processing interface that uses people to help authors to shorten, proofread, and edit documents.

This paper builds on the Find-Fix-Verify design pattern used in Soylent [3], where a different group of crowdworkers 1) Find errors in a given text (*Find*), 2) Fix them by editing (*Fix*), and finally 3) Verify the modifications (*Verify*).

3 USER INTERACTION

A group of people can use the *TourExplain* system when going on a trip. The group creates a new "Trip" in the system and enter trip parameters (i.e., POIs to be considered, number of participants, and whether the explanations need to be anonymous or not). Following the creation of the trip in the system, each member of the group has to enter their own preferences for each POI (in a private environment). After all of the preferences have been submitted, the system generates an itinerary, or a sequence of POIs, for the group, as well as explanations. Each explanation is then posted to the crowdsourcing system to be improved as described in Section 4.2 "Subsystem 2: Crowdsourcing". Once the crowdsourcing part of the system completes, each user will be received the recommended itinerary and its corresponding explanations.

4 SYSTEM DESIGN

Figure 1 outlines the workflow for the *TourExplain* system. It consists of two subsystems that communicate via an API: (1) *explanation* generation, and (2) *crowdsourcing* to improve the generated explanations. The implementation of our system supports the use of both subsystems, as well as the use of each individual module separately. Besides, this architecture allows us to easily add, exchange, or remove modules in our system.

4.1 Subsystem 1: Explanation

This subsystem consists of two parts: 1) *generate sequences*, and 2) *generate explanations.*

Generate sequences. Here, the system generates a sequence of POIs for the group to visit, according to previously proposed preference aggregation algorithms [12]. A preference aggregation strategy dictates how to combine individual preferences to recommend a sequence. This dictates both whether an item is included, as well as its position in the itinerary. The latter is important to consider since it has previously been found that overall satisfaction with a sequence depends on the order of the items in the sequence [11].

Following are the two above mentioned algorithms that we used to generate itineraries. Readers who wish to get a coherent overview of the proposed algorithms is referred to our previous work [12]:

- A 1: **Least Misery + Most Pleasure + Without Misery**. The plus signs imply chaining three strategies, applying one after the other.
- A 2: **Fairness -> Average**. The arrow implies applying a tie-breaking strategy, i.e., when several items receive an equal score using only Fairness.

Generate explanations. Pure crowdsourcing approaches to explain the preference aggregation strategies used to generate the sequence of recommended items will not succeed because most crowdworkers are not domain or recommendation experts. Even if they are informed about applied algorithms we cannot expect a crowdworker to write an appropriate explanation for the recommended items. Therefore, we provide them with initial explanations in the beginning which they can *improve* based on specific criteria.

Using a template-based natural language generation approach, the system generates explanations for each user according to their preferences in the recommended sequence. These (personal) explanations are based on predefined templates, examples of templates are *"Hello X"*, *"we know you would love to see Y"*, and *"however, others in your group would love to see Z"*.

For example, we consider a user *John* who has expressed a liking for seeing the Eiffel Tower because John and a couple of friends are visiting Paris soon. However, John's friends have expressed they preferred seeing the Louvre over the Eiffel Tower. This could lead to a template based sentence: *"Hello John, we know you would love to see the Eiffel Tower, however, others in your group would love to see the Louvre first."*

The system is provided with a number of templates to handle a number of predefined situations considered by the explanation generating algorithm. These automatically generated explanations are then sent to the second part of the system via an API to be reviewed by crowdworkers.

An example scenario for when an explanation may be needed is when a POI that is highly rated by person A is not chosen in the sequence of recommended POIs. The explanation for this person can be: *"Even though you wanted to visit POI X, most of your friends gave a very low rating for that POI. Therefore, we did not include that into the recommended POIs for the group."*

4.2 Subsystem 2: Crowdsourcing

The aim of the *crowdsourcing subsystem* is to improve the aforementioned generated explanations by using the wisdom of crowds.

We employ the Find-Fix-Verify pattern as described by [3] to detect and eradicate errors in the explanations. This approach not only flags up errors in explanations but also improves the explanations.

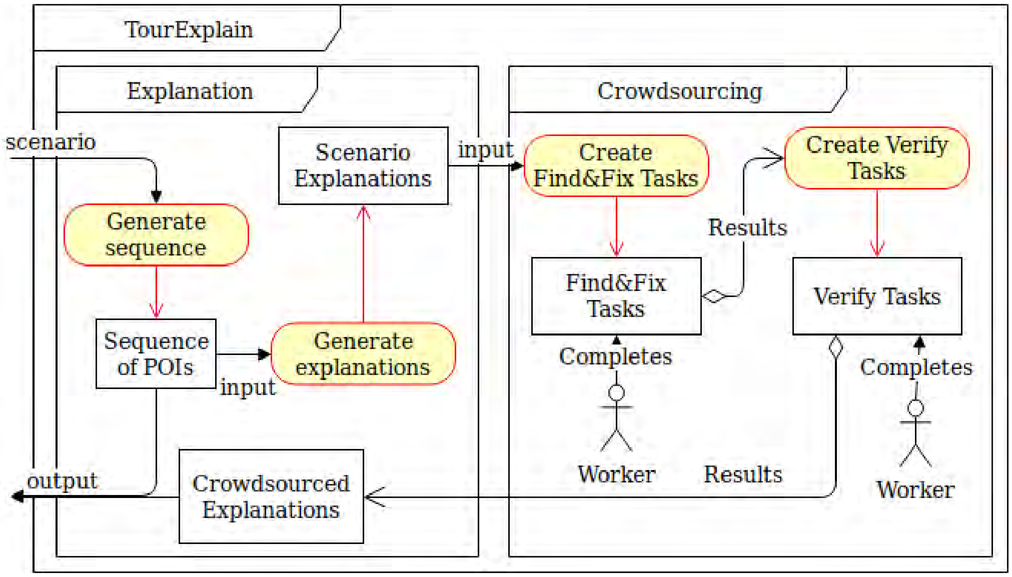


Figure 1: The system consists of two subsystems: 1) Explanation generation; 2) Crowdsourcing to improve the explanations.

For our purpose, we have adapted this approach and combined the Find- and Fix steps. This improves the accuracy of suggestions and is less time-consuming as well. Unlike the case of Soylent [3], the text to improve is short and can be modified efficiently. The same worker can directly suggest an improvement when they find an explanation inappropriate, as opposed to simply passing on that information to another worker, who then has to find an improvement.

Guidelines for Find-Fix-Verify. We give our workers three main criteria (based on Gricean Maxims [6]) to look for in the tasks:

- **Quantity:** Is the explanation informative? Does it provide all the information necessary and no more?
- **Quality:** Is the explanation truthful? Does it provide no information which is false?
- **Relevance:** Is the explanation relevant to the given scenario? It should not mention any irrelevant information.

The crowdsourcing pipeline contains two tasks:

Find & Fix tasks. A crowd-worker (worker henceforth) is given an explanation and asked to find any shortcomings in terms of the criteria mentioned above. After that, the worker is asked to make a suggestion to improve (fix) the sentence.

Verify task. A worker is given an explanation that is fixed by another worker in the find-fix step to evaluate in terms of the criteria mentioned above. The worker is asked to verify each criterion on a binary scale, giving their approval or disapproval for the particular metric. When the majority of the workers approve at least *two* criteria for a given explanation, it is considered a satisfactory explanation. Based on the number of approval/disapproval ratings, these satisfactory explanations are ranked from best to worst.

A vital part of our system is that the workers who do the *Find-Fix* versus *Verify* steps are independent of each other. This ensures there is no bias in picking a particular explanation. The tasks are created and launched using the Figure Eight API [3].

Unlike the previous generation of explanations, the crowdsourcing part cannot be done in real-time, but it requires some more time to be done. This is due to the fact that is not possible to know when the tasks will be performed by the workers. This time is subject to multiple factors as the monetary reward for each task, or the number of workers that perform the same task. In fact even though is possible to estimate the time to perform a given task by a worker, it becomes complex to estimate when a launched task will be picked up by a worker, also this time is directly related with the monetary reward for the task.

To ensure data quality, we only select workers that are native English speakers. When it was possible we randomized the order of questions and answers to avoid possible bias. Furthermore, to limit the introduction of error by the workers we performed each step by multiple workers. The number of workers that perform the same step can be dynamically chosen.

5 NEXT RESEARCH STEPS

We plan to use this pipeline as the basis of doctoral work investigating how to best generate explanations for itineraries (sequences of POIs) for groups of users. For this purpose, as suggested by Kim et al. [9], we are going to let crowdworkers form groups and collaborate to accomplish determined tasks.

In the following sections we describe future research avenues that will be pursued in this project. We introduce the notion of group dynamics, which consider the relationship between people within a group. We also consider the influence of interaction design on the requirements for explanations.

[3]https://www.figure-eight.com/, retrieved June 2018

5.1 Group Dynamics

Existing group recommendation techniques usually focus on merely aggregating individual preferences and thus do not take into account social interactions and relationships among the group members. Previous work has found that it is not the case, i.e., group members are influenced in their evaluations by the combination of the group and the interaction between and social relationships among group members [5].

In order to personalized the preference aggregation algorithms and their corresponding explanations as well as make our recommendations group-aware, we plan to use the Thomas-Kilmann Conflict Style Model (TKI model) [8] as a personality model. The advantage of this model is that it focuses on the interaction between group members rather than the characteristics of individual users, as in the Big Five factor model [5].

Another important group aspect that we aim to consider is the types of relationships within groups (c.f., [11]).

- Communal Sharing: Somebody you share everything with
- Authority Ranking: Somebody you respect highly
- Equality Matching: Somebody you are on equal footing with
- Market Pricing: Somebody you do deals with / compete with

We can employ the group types in both aggregating preferences algorithms as well as designing explanations. For instance you might feel comfortable to reveal your preferences to somebody you are on equal footing with (such as your friends) but not with somebody you respect (such as your boss).

5.2 Interaction Design

Individual vs Group Explanations. In this work we tried to improve automatically generated explanations by using the wisdom of crowds for a *single user*. However, we did not evaluate the final result with real groups of users. In our next steps, we will evaluate these explanations by presenting them to groups and compare the results with individual personalized explanations (for each group member). One can expect to find a trade-off between explanations that are suitable for the whole group, compared to personal explanations for each group member. For example, one benefit of group explanation is that we can present it on a common device viewed by the whole group. On the other hand, personal explanation can supply individual users with more personalized information about why that item is recommended to them.

Transparency vs Privacy Preserving. The requirements on explanations are also likely to be influenced by group versus individual preferences. For instance, there is a trade-off between having a high transparency while not violating the users' privacy. So users might demand to conceal their preferences for other group members or they feel comfortable to reveal their preference depends on different types of groups or their personalities.

Single Item vs Sequence Explanations. In this work we provided an explanation for each *single item*. However it might not be convenient always depending on several things e.g., domain. For example, in the cinema domain, users would want a recommendation for a specific movie instead of a sequence whereas in the tourism domain, a sequence of POIs would be more appropriate. In our future work, we will design explanations for each POI, and compare them with an explanation for the whole itinerary.

6 CONCLUSION

In this paper we introduce an automated crowdsourcing pipeline to generate and evaluate explanations for groups. The proposed solution is suitable for domains where items are a) consumed in groups, and b) in a sequence. This particularly useful for the recommendation of itineraries in tourism.

Additionally, it is likely that the approach is extendable to other domains, however there is a constraint for domains which require immediate and real-time explanations. For tourism, where trips can be planned in advance of a visit, this limitation may be less severe.

While simple, the proposed approach can be extended to answer different research questions. In this position paper we highlighted two significant and planned extensions:

- **Group dynamics**. How can explanations be improved by taking in account group dynamics such as conflict style or relationships within groups?
- **Interaction Design**. How should we adapt the explanations to the way they are consumed, e.g., for an individual item or for a sequence? Or for a single user versus for the group?

REFERENCES

[1] Liliana Ardissono, Anna Goy, Giovanna Petrone, Marino Segnan, and Pietro Torasso. 2003. Intrigue: personalized recommendation of tourist attractions for desktop and hand held devices. *Applied artificial intelligence* 17, 8-9 (2003), 687–714.

[2] Pablo J. Barrio, Daniel G. Goldstein, and Jake M. Hofman. 2016. Improving Comprehension of Numbers in the News. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2729–2739. https://doi.org/10.1145/2858036.2858510

[3] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: A Word Processor with a Crowd Inside. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 313–322. https://doi.org/10.1145/1866029.1866078

[4] Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. 2016. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 175–182.

[5] Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič. 2018. *Group Recommender Systems: An Introduction*. Springer.

[6] H Paul Grice. 1975. Logic and conversation. *1975* (1975), 41–58.

[7] Jeff Howe. 2006. Crowdsourcing: A definition. (2006).

[8] Ralph H Kilmann and Kenneth W Thomas. 1977. Developing a forced-choice measure of conflict-handling behavior: The "MODE" instrument. *Educational and psychological measurement* 37, 2 (1977), 309–325.

[9] Joy Kim, Sarah Sterman, Allegra Argent Beal Cohen, and Michael S. Bernstein. 2016. Mechanical Novel: Crowdsourcing Complex Work through Reflection and Revision. *CoRR* abs/1611.02682 (2016). arXiv:1611.02682

[10] Judith Masthoff. 2004. Group modeling: Selecting a sequence of television items to suit a group of viewers. In *Personalized digital television*. Springer, 93–141.

[11] Judith Masthoff. 2015. Group recommender systems: aggregation, satisfaction and group attributes. In *Recommender Systems Handbook*. Springer, 743–776.

[12] Shabnam Najafian and Nava Tintarev. 2018. Generating consensus explanations for group recommendations. In *UMAP Latebreaking results*.

[13] Thuy Ngoc Nguyen and Francesco Ricci. 2018. Situation-Dependent Combination of Long-Term and Session-Based Preferences in Group Recommendations: An Experimental Analysis. In *Symposium on Applied Computing (SAC)*. 1366–1373.

Challenges on Evaluating Venue Recommendation Approaches

Position paper

Pablo Sánchez
Universidad Autónoma de Madrid
Madrid, Spain
pablo.sanchezp@uam.es

Alejandro Bellogín
Universidad Autónoma de Madrid
Madrid, Spain
alejandro.bellogin@uam.es

ABSTRACT

Recommender systems are widely used tools in a large number of online applications due to their ability to learn the tastes and needs of the users. Venue recommendation approaches have recently become particularly useful, and even though these techniques have certain characteristics that differ from traditional recommendation, they deserve special attention from the research community due to the increase on the number of applications using tourism information to perform venue suggestions. In particular, how to properly evaluate (in an offline setting) this type of recommenders needs to be better analyzed, as they are normally evaluated using standard evaluation methodologies, neglecting their unique features. In this paper, we discuss and propose some solutions to two specific aspects around this problem: how to deal with already interacted venues in the test set and how to incorporate the sequence of visited venues by the user when measuring the performance of an algorithm (i.e., in an evaluation metric).

1 INTRODUCTION

The large development of location-based social networks (LBSNs) in recent years has encouraged research on the problem of Point-of-Interest (POI) or venue recommendation, i.e., suggesting new places for users to visit by analysing different contexts such as interaction patterns, friendship relationships, or geographical influence [13, 14]. Foursquare, Gowalla, or GeoLife, and many more, are examples of this kind of social networks, where users record check-ins they make to certain POIs (restaurants, cinemas, hotels, etc.) and share their opinions about them in the application [19, 20]. Because of this, many recommendation techniques have been proposed that exploit these information sources, see for example [7, 11, 12, 14, 19, 21]; however, a critical step to decide whether these algorithms are valuable or could be usable in the real world is the evaluation process, which should be realistic and performed with great care.

With this idea in mind, in this work we analyze some important aspects we have detected related to how POI recommendation tends to be evaluated in offline settings. Our driving hypothesis is that a recommender system should be evaluated in a situation as close as that where it would be used. Because of this, we consider that offline evaluation should be performed by running a temporal split, where the recommender should predict the present (or future) user interactions based on her past interactions [5]. However, independently of whether a temporal split was used, we have detected two challenges that shall be the focus of this paper: first, how should we deal with those venues the user already visited in the past?, and second, can we incorporate the actual order followed by the user (in the test set) to assess the accuracy of the provided recommendations?

In the next sections we motivate and present these two challenges in more detail, and present later some preliminary experiments we have obtained, together with some conclusions and future ideas related to these issues.

2 EVALUATION METHODOLOGIES: KNOWN VS NEW VENUES

In classical recommender systems, no repetitions are typically allowed or considered in the datasets, probably inherited by the domains of the first available datasets (movies) [8, 9]; however, in the venue recommendation context users often visit the same place more than once, and hence, it may make sense to consider how these repetitions should be incorporated in the models and in the evaluation process. This behavior is not limited to venue recommendation, it also happens in music or e-commerce recommendation, and tasks such as session-based recommendation or automatic playlist continuation [16]. Nonetheless, to the best of our knowledge, there is no thorough research about the effects of this paradigm shift, especially regarding the evaluation of the recommendation techniques.

Some papers explicitly state that they separate venues, instead of check-ins, hence, in those cases it is clear that there are no known or visited items in the test set by that user (see [12–14]). However, in other situations it is not obvious how the test set is created, for instance, when temporal splits are created, where repetitions may naturally occur and it is not clear if already known venues were removed from the test set of the user [20]. As we shall see in our experiments, these experimental settings may have a profound impact on the performance of the recommenders and on the observed trends, not only from a reproducibility perspective; hence, the community would benefit from a careful analysis about this issue.

We argue that, by evaluating with items already interacted by the user we are aiming at a different kind of algorithm than when those items are removed. In other terms, a recommender system that performs very well in the first scenario (with known items) is expected to distinguish well which of the previously visited venues the user will visit next. In this context, its final goal is to generate recommendations already known by the user, probably the opposite of a recommender evaluated with only new items in the test set, thus aiming at recommending new, novel venues for each particular user – in fact, some authors define explicitly such a task as *recommending new places* [4].

Our assumption – that we would like to study in the future, since it is out of the scope of this position paper – is that those recommenders that better predict in the case of known venues, would probably generate less novel recommendations in general.

Table 1: Description of the temporal partition evaluated created based on the Foursquare dataset, where U, I, and C denote the number of users, items, and check-ins.

Check-in period	U	I	C	Density	C/U	C/I
Apr'12-Sep'13	267k	3.6M	33M	0.0034%	123.596	9.16
Training: May-Oct '12	202k	1.1M	4.7M	0.0021%	23.267	4.278
Test: Nov '12	150k	352k	831k	0.0017%	5.540	2.361

We believe it would be interesting to understand this effect, in part, to improve current recommendation algorithms that perform well in either of these tasks by creating a hybrid algorithm useful in a real use-case scenario, in such a way that it would detect if new or already visited recommendations should be returned to a user, based on her previous interactions.

3 INTEGRATING SEQUENCES IN EVALUATION

Another specific feature of venue recommendation that differs from the more traditional recommendation problem is that the order in which users visit the venues provides a lot of information. Recently, some methods have been proposed that provide recommendations based on temporal or sequential aspects, such as [6, 21]. However, this information has been neglected, so far, when evaluating these algorithms. Except for the work presented in [6], where the authors propose a metric based on F1 that takes into account the pairwise order between POIs, we have not found other approaches where the evaluation metrics explicitly compare the order of the recommendations against the visited venues.

Furthermore, and related to the discussion presented in the previous section, classical ranking metrics fit the scenario with no repeated items, however, they cannot be adapted to the case where repetitions exist (at least, not to the case where there are repetitions in the test set). Because of this, we believe sequences should be formally integrated and considered when evaluating recommender systems in the venue recommendation context.

With this idea in mind, herein we propose an evaluation metric based on the Longest Common Subsequence (LCS) algorithm, a technique used to find a subsequence of elements (no necessary consecutive) whose length is the maximum possible between two sequences [1]. In our context, one of the sequences will be the recommendation list (R_u) and the other the actual visited venues that appear in the test set of the user (T_u, ordered by ascending timestamp); in this way, the LCS algorithm will measure how many items were recommended in the same order as the user visited them. For instance, if the sequence of items ABCDE is found in the test set of a user, and one recommender suggests ABXCD, whereas another provides ABDXC, the LCS algorithm will score higher the first one, since the subsequence found (exploiting not consecutive items) is larger in that case (4 against 3).

Finally, since the LCS between two sequences is not bounded, we need to normalize this value (lcs). We propose the following three variations when measuring rankings at cutoff N of both recommended and test sequences: $LCSP(R_u, T_u) = lcs(R_u, T_u)/N$ (based on precision), $LCSR(R_u, T_u) = lcs(R_u, T_u)/|T_u|$ (based on recall), and $LCS(R_u, T_u) = lcs(R_u, T_u)^2/(N \cdot |R_u|)$.

4 PRELIMINARY EXPERIMENTS

The experiments have been performed using the global-scale check-in dataset of Foursquare[1] made public by the authors of [17, 18]. Starting from more than 33M check-ins, we created one temporal split containing 6 months of data in its training split and one month for testing (more statistics are shown in Table 1). As a pre-processing step, we performed a 2-core before splitting the data into training and test, so that we force that every user and item has at least 2 check-ins.

We report results obtained by the following recommenders:

- Random (Rnd): random recommender.
- Popularity (Pop): recommender that suggests the most popular items, i.e., items with more check-ins.
- AvgDis: baseline that recommends the closest POIs to the user's average location. The average is computed by calculating the midpoint of the coordinates of the POIs visited by the user.
- PGN: a hybrid approach similar to the USG model proposed in [19] that combines a user-based method (UB), Pop, and AvgDis recommenders. It basically aggregates the scores of every item provided by each of the recommenders, after normalizing each score by the maximum score of each method.
- UB: a k-NN recommender with a user-based approach [15].
- IB: a k-NN recommender with an item-based approach [15].
- HKV: a matrix factorization (MF) approach as described in [10] that uses Alternate Least Squares in the minimization formula.
- IRenMF: weighted MF method proposed by [14]. We selected this approach because, according to the comparison presented in [13], IRenMF was very competitive with a lower execution time with respect to other models, such as GeoMF, Rank-GeoFM, or LFBCA, which agrees with some preliminary experiments we performed in our dataset.

Based on the temporal split presented in Table 1, we decided to focus on the 2 largest cities in terms of number of check-ins (Jakarta and Istanbul) and create 2 independent training-test datasets. Furthermore, in order to make a fair comparison among all the evaluated baselines, we removed repetitions in a user basis for the classical collaborative filtering algorithms; we kept two versions of the training set (with and without check-in frequencies) so that some POI recommendation algorithms, in our case AvgDis and IRenMF, could exploit the frequency of users when visiting a specific venue (denoted as AvgDisFreq and IRenMFFreq). Additionally, to test the experimental conditions discussed in Section 2, we created two test sets: one where those venues the user already interacted in the past (training set) are removed (*with new venues*) and another where they are kept (*with known venues*).

To evaluate the recommenders under the *with known venues* strategy we selected as candidates for each user all the venues that appear in the complete training set of each target city, while when working *with new venues* we remove the ones already rated by that user.

[1] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

We use different ranking metrics to measure accuracy of the recommenders: precision (P), recall (R), mean average precision (MAP), and normalized discounted cumulative gain (NDCG) [2]. We also report the proposed metrics based on LCS, as presented in Section 3. The parameters of the recommenders have been selected by maximizing P@5. Unless stated otherwise, the reported values are computed at a cutoff of 10. Source code to replicate these experiments can be found in the following Bitbucket repository: PabloSanchezP/TempCDSeqEval.

4.1 Comparison of evaluation methodologies

Table 2 shows the results for the cities mentioned before evaluated under the two methodologies presented in Section 2: where only new items for a user appear in her test set (*with new venues*) and where venues already interacted by the user are allowed in the test set (*with known venues*). Nevertheless, for the test set, we always removed the duplicated check-ins (i.e., the users only made one check-in in a POI). As a simple baseline, we have included a method that returns the venues observed in training for each user (Training), ordered by their score and popularity. This baseline, as expected, does not obtain any relevant result in the first scenario, however, when known items are allowed, it is a strong baseline to beat, and some of the more complex algorithms such as IRenMF tend to obtain performance values very close to the ones from this method.

We also notice that in the *with new venues* scenario, the well performing methods tend to be very close to each other (see PGN, UB, IRenMF, and IRenMFFreq in Jakarta), however, in the other scenario the differences increase and some methods take more advantage than others of the different experimental condition.

Another interesting observation is that, as already happens in classical recommendation [3], a popularity bias is found when evaluating in the *with new venues* scenario; however, this bias is strongly reduced in the *with known venues* scenario, favoring the Training baseline, evidencing that in such scenario well-known, popular venues are not as important as previously visited venues by each user, confirming that these two scenarios are actually modeling two different recommendation situations and hypotheses.

4.2 Sequence-aware evaluation metric

To test the evaluation metric proposed in Section 3 based on the LCS algorithm, in Table 2 we have included two methods as skylines (named like this as opposed to the baselines, since their performance is almost impossible to achieve because they look into the test set): TestOrder, that returns the test set in the (ideal) observed order visited by the user (from lowest to highest timestamp), and TestInvOrder, that also returns the test but in the reverse order (from highest to lowest timestamp). The use of these recommenders will serve to justify the LCS-based metric, since besides taking into account the relevance, it also considers the order of visits (note that none of the other recommenders explicitly generates sequences of items, we aim to address this issue in the future). We observe that the LCS-based metrics (LCS, LCSP, LCSR) produce lower values for TestInvOrder than for TestOrder, as TestInvOrder only finds one item in the correct sequence when using these metrics; however, since TestInvOrder obtains much better results than traditional recommenders, we conclude that, for many users, the other algorithms are not able to obtain a single relevant item. At the same time, the skylines obtain the same values by any of the other ranking-based metrics (P, R, MAP, NDCG) since they do not consider the visiting order of the recommended list.

Based on these results, we can provide additional insights about how the different recommendation algorithms behave. For instance, in the *with known venues* scenario, the Training baseline seems to provide recommendations more often in the same order as the one observed in the test set, since the values for the LCS metric is always higher than for any of the sequence-agnostic evaluation metrics. In the other scenario, on the other hand, we do not observe too many variations on how the recommenders are being ranked by each evaluation metric, hence, further analysis and experiments should be performed to better understand this effect. One possible reason for this lack of variability in the performance could be related to the very small number of relevant items returned by the algorithms, in the future we would like to study this problem in more detail.

5 CONCLUSIONS AND FUTURE WORK

In this work, we discuss two aspects regarding how the community should address the evaluation of venue recommendation approaches. First, we analyze whether repeated interactions should be included in the test splits, observing how state-of-the-art recommendation algorithms change under these different experimental conditions. Considering this type of behavior is common in the tourism domain – and inherent to some type of tourists – the presented observations could open up for discussion about how this issue should be addressed in the community, especially, which scenario is more interesting from an offline point-of-view of the evaluation process, without forgetting that some recommendations might be obvious (hence, less useful) for the users [4], as evidenced by the good performance achieved when returning those venues already visited by the user. We aim to continue investigating about this problem in the future, especially about the connection between (lack of) novelty and observed accuracy under experimental conditions with known items. An important issue we aim to address is the best way to exploit check-in datasets such as the one used here, since there is no difference between tourists and locals (which may check-in in nearby places or visit locations as part of their daily life) and, hence, we want (and need) to understand if the derived conclusions concern to locals, tourists, or both.

The second aspect we have presented here is related to the use of sequences in the evaluation of POI recommendation approaches. We have defined an evaluation metric based on the Longest Common Subsequence that takes into account how similar the recommended list is with respect to the order the user checked in the venues. In the future, we would like to explore how this metric behaves on different tasks related to tourism recommendation, such as next-POI recommendation and tour recommendation, where the recommendation order plays an important role. Furthermore, we aim to incorporate in our analysis algorithms that explicitly recommend sequences of items [16].

ACKNOWLEDGMENTS

This work was funded by the project TIN2016-80630-P (MINECO).

Table 2: Performance comparison on 2 different cities including already interacted items by the user in the test set (*with known venues*) and excluding such items (*with new venues*). Our proposal for a sequence-aware evaluation metric is also included (LCS, LCSP, LCSR). Best results are denoted in bold: with † when TestInvOrder and TestOrder are not considered, without † when also the baselines (Rnd, Pop, Training) are not considered.

(a) Istanbul

Recommender	Test with new venues							Test with known venues						
	P	R	NDCG	MAP	LCS	LCSP	LCSR	P	R	NDCG	MAP	LCS	LCSP	LCSR
Rnd	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Pop	0.039	0.076	0.063	0.030	0.008	0.034	0.071	0.054	0.082	0.079	0.036	0.009	0.046	0.075
Training	0.000	0.000	0.000	0.000	0.000	0.000	0.000	†**0.120**	†**0.190**	0.186	0.100	†**0.034**	0.090	†**0.157**
AvgDis	0.001	0.001	0.001	0.001	0.000	0.001	0.001	0.003	0.006	0.007	0.005	0.001	0.002	0.006
AvgDisFreq	0.001	0.002	0.001	0.001	0.000	0.001	0.001	0.003	0.007	0.008	0.005	0.001	0.003	0.006
PGN	0.041	0.082	0.073	0.036	0.009	0.037	0.077	0.070	0.112	0.124	0.065	0.013	0.059	0.101
UB	0.045	0.088	0.078	0.039	0.009	0.039	0.081	0.110	0.167	0.178	0.098	0.021	0.086	0.142
IB	0.036	0.069	0.063	0.032	0.008	0.032	0.064	0.108	0.156	0.175	0.098	0.019	0.082	0.130
HKV	0.043	0.087	0.076	0.039	0.009	0.038	0.080	0.105	0.158	0.170	0.093	0.019	0.082	0.135
IRenMF	0.044	0.089	0.077	0.039	0.010	0.039	0.083	0.100	0.151	0.164	0.090	0.018	0.079	0.130
IRenMFFreq	†**0.047**	†**0.094**	†**0.082**	†**0.042**	†**0.010**	†**0.041**	†**0.087**	**0.117**	**0.181**	†**0.194**	†**0.109**	**0.023**	†**0.092**	**0.154**
TestInvOrder	0.468	0.932	0.978	0.967	0.225	0.100	0.356	0.569	0.910	0.985	0.978	0.162	0.100	0.287
TestOrder	0.468	0.932	0.978	0.967	0.932	0.468	0.932	0.569	0.910	0.985	0.978	0.910	0.569	0.910

(b) Jakarta

Recommender	Test with new venues							Test with known venues						
	P	R	NDCG	MAP	LCS	LCSP	LCSR	P	R	NDCG	MAP	LCS	LCSP	LCSR
Rnd	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Pop	0.029	0.076	0.070	0.044	0.008	0.026	0.073	0.044	0.087	0.091	0.056	0.009	0.038	0.082
Training	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.102	0.196	0.171	0.096	†**0.034**	0.078	0.165
AvgDis	0.001	0.002	0.002	0.001	0.000	0.001	0.002	0.003	0.008	0.007	0.005	0.001	0.002	0.007
AvgDisFreq	0.001	0.002	0.001	0.001	0.000	0.001	0.002	0.004	0.010	0.009	0.006	0.001	0.003	0.009
PGN	0.030	0.078	0.072	†**0.045**	0.008	0.027	0.075	0.056	0.108	0.114	0.069	0.012	0.047	0.100
UB	0.036	0.085	0.075	0.043	0.009	0.032	0.081	0.081	0.141	0.146	0.083	0.019	0.065	0.124
IB	0.019	0.045	0.038	0.021	0.005	0.017	0.043	†**0.120**	†**0.212**	†**0.222**	†**0.141**	**0.026**	†**0.088**	†**0.172**
HKV	0.035	0.084	0.071	0.039	0.009	0.032	0.080	0.078	0.137	0.138	0.078	0.016	0.063	0.121
IRenMF	0.033	0.081	0.071	0.041	0.009	0.030	0.078	0.076	0.135	0.136	0.078	0.016	0.062	0.121
IRenMFFreq	†**0.036**	†**0.092**	†**0.077**	0.044	†**0.010**	†**0.033**	†**0.088**	0.110	0.199	0.193	0.115	0.024	0.084	0.170
TestInvOrder	0.387	0.923	0.963	0.947	0.299	0.100	0.427	0.492	0.912	0.977	0.966	0.223	0.100	0.348
TestOrder	0.387	0.923	0.963	0.947	0.923	0.387	0.923	0.492	0.912	0.977	0.966	0.912	0.492	0.912

REFERENCES

[1] Alberto Apostolico. 1997. String Editing and Longest Common Subsequences. In *Handbook of Formal Languages 2*, Grzegorz Rozenberg and Arto Salomaa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 361–398.

[2] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 2011. *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England.

[3] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2017. Statistical biases in Information Retrieval metrics for recommender systems. *Inf. Retr. Journal* 20, 6 (2017), 606–634.

[4] Cécile Bothorel, Neal Lathia, Romain Picot-Clémente, and Anastasios Noulas. 2018. Location Recommendation with Social Media Data. In *Social Information Access*. Lecture Notes in Computer Science, Vol. 10100. Springer, 624–653.

[5] Pedro G. Campos, Fernando Díez, and Iván Cantador. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Model. User-Adapt. Interact.* 24, 1-2 (2014), 67–119.

[6] Dawei Chen, Cheng Soon Ong, and Lexing Xie. 2016. Learning Points and Routes to Recommend Trajectories. In *CIKM*. ACM, 2227–2232.

[7] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*. ACM, 93–100.

[8] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*. Springer, 265–308.

[9] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *TiiS* 5, 4 (2016), 19:1–19:19.

[10] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. IEEE Computer Society, 263–272.

[11] Xutao Li, Gao Cong, Xiaoli Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-GeoFM: A Ranking based Geographical Factorization Method for Point of Interest Recommendation. In *SIGIR*. ACM, 433–442.

[12] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *KDD*. ACM, 831–840.

[13] Yiding Liu, Tuan-Anh Pham, Gao Cong, and Quan Yuan. 2017. An Experimental Evaluation of Point-of-interest Recommendation in Location-based Social Networks. *PVLDB* 10, 10 (2017), 1010–1021.

[14] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting Geographical Neighborhood Characteristics for Location Recommendation. In *CIKM*. ACM, 739–748.

[15] Xia Ning, Christian Desrosiers, and George Karypis. 2015. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In *Recommender Systems Handbook*. Springer, 37–76.

[16] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* 51, 4 (2018), 66:1–66:36.

[17] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. 2015. NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *J. Network and Computer Applications* 55 (2015), 170–180.

[18] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory Cultural Mapping Based on Collective Behavior Data in Location-Based Social Networks. *ACM TIST* 7, 3 (2016), 30:1–30:23.

[19] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*. ACM, 325–334.

[20] Jia-Dong Zhang and Chi-Yin Chow. 2013. iGSLR: personalized geo-social location recommendation: a kernel density estimation approach. In *SIGSPATIAL/GIS*. ACM, 324–333.

[21] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. 2014. LORE: exploiting sequential influence for location recommendations. In *SIGSPATIAL/GIS*. ACM, 103–112.

