RecTour 2016

Workshop on Recommenders in Tourism

Boston, MA, USA, September 15th, 2016

Proceedings

Edided by Daniel Fesenmaier, Tsvi Kuflik,
and Julia Neidhardt

Co-located with the 10th ACM Conference on
Recommender Systems (RecSys 2016)

The ACM Conference Series on
Recommender Systems

Preface

This volume contains the contributions presented at the Workshop on Recommenders in Tourism (RecTour), hold in conjunction with the 10th ACM Conference on Recommender System (RecSys 2016), in Boston, MA, USA. The proceedings are also published online by CEUR Workshop Proceedings at http://ceur-ws.org/Vol-1685/.

RecTour 2016 focuses on the specific challenges for recommender systems in the tourism domain. In this domain, there are considerably more complicated scenarios than finding the best product for a user. Planning a vacation usually involves searching for a set of products that are interconnected (e.g. means of transportation, lodging, attractions etc.), with a rather limited availability, and where contextual aspects may have a major impact (spatiotemporal context, social context, environmental context). In addition and most importantly, products are emotionally "loaded" and therefore decision taking is not based on rational and objective criteria (i.e., system 2 thinking). As such, providing the right information to visitors of a tourism site at the right time about the site itself and various services nearby is challenging. Additionally and in contrast to many other domains, information providers are normally SMEs and do not have full information about available opportunities. Moreover, there is no single, standard format to house this information. Thus, given this diversity, building effective recommendation 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 travellers no longer rely on travel agents/agencies. Indeed, recent studies indicate that they are now active in searching for information and composing their vacation packages according to their specific preferences. When onsite, 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the site itself rather than renting a visitor guide that may be available, but considered to be expensive and sometimes outdated.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. Recommender systems were suggested as a practical tool for overcoming this information overload. However, the tourism domain is substantially more complicated, and as such, creates huge challenges for those designing tourism focused recommender systems.

The workshop aims at bringing together researchers and practitioners working in the tourism recommendation domain, in order to look at the challenges from the point of view of the user interactions as well as from the point of view of service providers and from the points of view of additional stakeholders as well (destination management organizations for instance). All in all, the workshop aims at attracting presentations of novel ideas for addressing these challenges and how to advance the current state of the art in this field. The primary goal of this workshop is to provide a forum for researchers and practitioners from different fields, e.g., tourism, recommender systems, user modelling, user interaction, mobile, ubiquitous and ambient technologies, artificial intelligence and web information systems, to explore various practical use cases of applications of these technologies in tourist recommender systems of the future. During the workshop we aim to identify the typical user groups, tasks and roles in order to achieve an adequate personalization and recommendation for tourism applications.

September 2016 — Daniel Fesenmaier, Tsvi Kuflik and Julia Neidhardt

Workshop Committees

Organizers

- Daniel Fesenmaier, University of Florida, USA
- Tsvi Kuflik, The University of Haifa, Israel
- Julia Neidhardt, TU Wien, Austria

Program Committee

- Ulrike Gretzel, The University of Queensland, Australia
- Antonio Moreno, Universitat Rovira i Virgili, Spain
- Francesco Ricci, University of Bozen/Bolzano, Italy
- Hannes Werthner, TU Wien, Austria
- Wolfgang Wörndl, Technische Universität München, Germany
- Zheng Xiang, Virginia Tech, USA
- Markus Zanker, University of Bozen/Bolzano, Italy

Workshop Program

9:00 - 10:30 Session 1

9:00 - 9:15 Workshop opening and motivation

9:15 - 10:00 Keynote *E-tourism: History and Challenges* by Hannes Werthner

10:00 - 10:30 Research papers: Context-aware recommender systems I

• Mesut Kaya and Derek Bridge: *Improved Recommendation of Photo-Taking Locations using Virtual Ratings.*

10:30 - 11:00 Coffee break

11:00 - 12:30 Session 2

11:00 - 12:30: Research papers: Context-aware recommender systems II

• Patrick Hiesel, Matthias Braunhofer and Wolfgang Wörndl: *Learning the Popularity of Items for Mobile Tourist Guides.*

• Christoph Trattner, Alexander Oberegger, Lukas Eberhard, Denis Parra and Leandro Balby Marinho: *Understanding the Impact of Weather for POI Recommendations.*

• Khadija Vakeel and Sanjog Ray: *A Motivation-Aware Approach for Point of Interest Recommendations.*

12:30 - 14:00 Lunch break

14:00 - 15:30 Session 3

14:00 - 15:00 Research papers: Advanced topics in tourism recommender systems

• Amra Delic, Julia Neidhardt, Thuy Ngoc Nguyen and Francesco Ricci: *Research Methods for Group Recommender Systems.*

• Paula Saavedra, Pablo Barreiro, Roi Durán, Rosa Crujeiras, María Loureiro and Eduardo Sánchez Vila: *Choice-based recommender systems.*

15:00 - 15:30 Demo papers: Event recommendations

• Stacey Donohue, Nevena Dragovic and Maria Soledad Pera: *Anything Fun Going On? A Simple Wizard to Avoid the Cold-Start Problem for Event Recommenders.*

• Sean MacLachlan, Nevena Dragovic, Stacey Donohue and Maria Soledad Pera: *"One Size Doesn't Fit All": Helping Users Find Events from Multiple Perspectives.*

15:30 - 16:00 Demo session/Coffee break

16:00 - 17:30 Session 4

16:00 - 16:45 Position papers: Further research challenges

• Jan Fabian Ehmke, Dirk Christian Mattfeld and Linda Albrecht: *Combining Mobility Services by Customer-Induced Orchestration.*

• Daniel Herzog and Wolfgang Wörndl: *Exploiting Item Dependencies to Improve Tourist Trip Recommendations.*

• Manoj Reddy Dareddy: *Challenges in Recommender Systems for Tourism.*

16:45 - 17:30 Panel discussion and workshop summary

Panel discussion: *Specific challenges for recommender systems in the tourism domain.*

• Daniel Fesenmaier, University of Florida, USA

• Hannes Werthner, TU Wien, Austria

• Wolfgang Wörndl, Technische Universität München, Germany

Table of Contents

Research Papers

Demo Papers

Position Papers

Improved Recommendation of Photo-Taking Locations using Virtual Ratings

Mesut Kaya
Insight Centre for Data Analytics
University College Cork, Ireland
mesut.kaya@insight-centre.org

Derek Bridge
Insight Centre for Data Analytics
University College Cork, Ireland
derek.bridge@insight-centre.org

ABSTRACT

We consider the task of collaborative recommendation of photo-taking locations. We use datasets of geotagged photos. We map their locations to a location grid using a geohashing algorithm, resulting in a *user* $\times$ *location* implicit feedback matrix. Our improvements relative to previous work are twofold. First, we create virtual ratings by spreading users' preferences to neighbouring grid locations. This makes the assumption that users have some preference for locations close to the ones in which they take their photos. These virtual ratings help overcome the discrete nature of the geohashing. Second, we normalize the implicit frequency-based ratings to a 1-5 scale using a method that has been found to be useful in music recommendation algorithms. We demonstrate the advantages of our approach with new experiments that show large increases in hit rate and related metrics.

CCS Concepts

•Information systems $\rightarrow$ Recommender systems;

Keywords

photography; geohashing; implicit ratings

1. INTRODUCTION

The advent of digital cameras and smart phones has revolutionised photo-taking. We now create more multimedia content than ever before; the content is more heavily contextualised, e.g. with increasingly accurate data from phone sensors such GPS receivers; and sharing the content has never been easier nor more commonplace.

Web sites, such as Flickr and Facebook, where multimedia content is shared, implicitly capture contextualised personal preferences over the places that people like to create content of this kind, i.e. the places where they take photos. The preference data is a resource from which we can build personalized and contextualized recommender systems [10].

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

In this paper, we study the task of recommending geolocations to users, based on the data found in photo-sharing sites. In other words, the *items* in our recommender system are geolocations. The recommender may help a user discover new places where they can enjoy good views or nice settings, suitable for photo-taking.

Collaborative recommendation of geolocations for photo-taking has been explored in recent work by Phan et al.[13] Briefly, they use a cartographic hashing function to map the latitude and longitude coordinates associated with photos to rectangular bins: photos taken from within the same bin have the same hash key. The *user* $\times$ *location* implicit feedback matrix uses the hash keys for *location*s. The rating by a user for a location is given by the proportion of her geotagged photos whose coordinates map to that bin; ratings are therefore in $(0, 1]$. Phan et al. compared an item-based nearest-neighbours recommender with three matrix factorization methods. They ran experiments measuring RMSE, with non-negative matrix factorization having lowest RMSE.

In this paper, we, like Phan et al., are concerned with recommending locations. We use geohashing rather than Phan et al.'s cartographic hashing to map the places where users took the photos to buckets. Then we make two main contributions. First, we create virtual ratings by spreading users' preferences to neighbouring grid locations. This makes the assumption that users have some preference for locations close to the ones in which they take their photos. These virtual ratings help overcome the discrete nature of the geohashing. Second, we normalize the implicit frequency-based ratings to a 1-5 scale using a method that has been found to be useful in music recommendation algorithms. We evaluate the effect of these two innovations separately and together using experiments that measure hit rate and related metrics, rather than RMSE.

In Section 2 we review the related work; in Section 3 we present our proposed method; and in Section 4 we give the experimental results.

2. RELATED WORK

There is an amount of previous research in recommending locations to users, e.g. [11, 3, 16, 21]. For the most part, this work is concerned with point-of-interest (POI) recommendation. For photo-taking, by contrast, we are not directly interested in recommending POI locations; instead, we want to be able to recommend locations that may give views of POIs (and nice settings for other photos). The locations that we recommend may even be far from any POIs. Hence, following [13], we recommend rectangular cells in the coordinate space. Phan et al. map latitude and longitude coordinates to rectangular bins using a method of their own invention, which they call Cartographic Sparse Hashing (CASH) [1]. Their method has a parameter, r, the resolution. At the Equator, bins will be r metres wide and r metres high. Note, however, that bins will be taller than r metres the further they are away from the Equator due to the curvature of the Earth. The resulting hash key is a 64-bit integer whose high bits are the hash of the longitude and whose low bits are the hash of the latitude. In more recent work, they use CASH within an activity recommender [1].

There are other location recommenders that also work in a coordinate space. For example, Liu et al. try to predict the next location that a user will visit [12]. Interestingly, they, along with Yuan et al. [21], also consider the role of time in location recommendation, which may also be relevant to photo-taking, but which we do not investigate further here.

Shared photos have been used as a data source for purposes such as POI detection [20], tag recommendation [17], photo-taking location detection [5], and route recommendation [14]. Some work specifically uses Flickr data, just as we and Phan et al. use in our work; for example, Zheng et al. recommend Flickr interest groups to users [22]. But none of this work, other than Phan et al.'s, uses this kind of data to recommend photo-taking locations.

The literature also contains descriptions of systems that assist with photo composition, e.g. [2, 15]. Bourke et al. describe what they call the *social camera*, which recommends a list of popular photos that were taken near to the user's location in similar lighting conditions. The user can choose one of these recommended photos, which will then be used as the basis for assistance with camera settings and framing [2]. Rawat proposes a system called ClickSmart that can provide real-time advice about scene composition and camera settings using rules learned from social media images [15].

Our use of virtual ratings is similar in spirit to the approach of fuzzy event modelling proposed by Hidasi and Tikk [7]. They use a similar idea to model continuous contexts in factorization algorithms.

3. PROPOSED METHOD

In this section, we explain our proposed approach. The approach consists of geohashing, followed by the spreading of users' preferences by creating virtual ratings in neighbouring buckets, followed by the conversion of implicit feedback to 1-5 ratings. Finally, we use a collaborative recommendation algorithm on the resulting feedback matrix.

3.1 Geohashing

We do not use Phan et al.'s CASH method, preferring to use *geohashing*, which is more common. Both have the same effect: they divide the surface of the Earth into a grid of rectangular cells (called bins in CASH and buckets in geohashing); a hash function takes in latitude and longitude and maps them to one of the cells of the grid. Geohashing maps latitude and longitude into a geohash key of up to 12-characters. Coordinates that map to the same key are in the same cell of the grid (bucket). The scheme is hierarchical: prefixes of the hash key designate larger cells that include those designated by extensions of the prefix. The size of the prefixes is known as the *precision*. In this paper we take 7-character long prefixes. These designate buckets that at the Equator are 152.9 metres wide and 152.4 metres high.

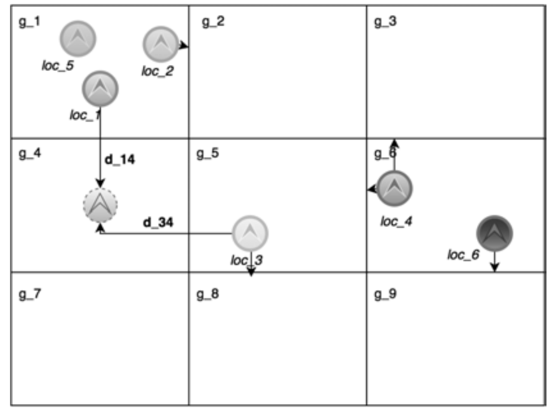


Figure 1: Geohashing

It is interesting to note that Phan et al. run experiments where they vary the resolution of their hashing method. Altering the resolution, however, does not just affect sparsity, it also alters the items (bins) that get recommended. Arguably, this should not be a parameter that one alters to minimise error. It is instead something one should fix at a granularity that users find useful for photo-taking. As we said, we fix precision at 7, which give buckets that are about 150×150 metres, which we think is an appropriate size for photo-taking location recommendation. Had we used a precision of 6, buckets would be approximately 1km by 600 metres, which is clearly too large for useful recommendations. A case can be made for precision of 8 (about 40 by 20 metres) but anything higher is probably too small (e.g. precision of 9 recommends 5 by 5 metre locations).

It is important also to say that the size of the buckets, which we use to recommend *where* to take photos, is unrelated to the size of *what* might be photographed. From a bucket on the south bank of the River Thames, for example, a user might capture a panoramic shot of the north London skyline or she might zoom in on a pigeon eating a discarded hamburger. This emphasises the point too that recommending photo-taking buckets is not the same as recommending POIs. In the same example of a bucket on the south bank of the River Thames, there might be a POI in the same bucket (e.g. the London Eye) but the user might be taking photos of POIs in a different bucket in the distance (e.g. Big Ben) or may not be taking photos of specific POIs at all (e.g. skylines and pigeons).

After hashing, we have an initial *user* $\times$ *location* ratings matrix, where locations are buckets and ratings are based on frequencies. Figure 1 shows an example. Suppose a user u has taken six photos in six different locations. Suppose loc_1, loc_2 and loc_5 are geohashed to the same bucket g_1. The ratings matrix contains triples such as $\langle u, g_1, 3\rangle$, meaning that user u has taken three photos in bucket g_1.

3.2 Creating virtual ratings

One problem with hashing to a rectangular grid is its discretization of coordinate space. In Figure 1, for example, taking a photo at loc_2 is taken as positive feedback for that point in space and others near it. But the rating is recorded only for bucket g_1. The geohashing results in us recording no positive feedback for the nearby points in g_2.

Our solution to this problem is to create virtual ratings in the $user \times location$ matrix by spreading the original frequencies to neighbouring buckets. First, we decide which buckets to spread to. We may spread to zero, one or more of the eight neighbouring buckets. We only spread from a bucket to a neighbour if the bucket contains a photo-taking event that is close enough to the neighbour. We calculate the geodesic distance between the coordinates of the photo-taking events in the bucket and the centre of the neighbour.[1] Only if the minimum of these distances is smaller than a threshold value Δ will we create a virtual rating. For example, in Figure 1, the rating for g_1 will only be spread to g_4 if the distance between loc_1 and the centre of g_4 (this being smaller than the distances from loc_2 and loc_5) is smaller than Δ.

Next, we decide the value of the virtual rating. Its value is a discounted version of the one that is being spread. Following [21], we use a power law distribution to model the preference of a user for a neighbouring bucket as a function of the minimum distance we calculated previously. This maps a distance of 0 to a weight of 1.0 and it maps the maximum distance (Δ) to a weight of 0.0. The neighbour's virtual rating is the product of the weight and rating (frequency) associated with the source bucket.

There is, however, the issue of how to aggregate ratings that 'arrive' in a bucket from different sources. For such cases, we use the simple heuristic that the virtual rating is the *maximum* of the ratings arriving from different sources. For example, in Figure 1, bucket g_4 receives two virtual ratings. One comes from g_1: it is g_1's rating (3) discounted by an amount based on the distance d_{14}. The other comes from g_5: it is g_5's rating (1) discounted by an amount based on distance d_{34}. The larger of these will be taken as the virtual rating for g_4. Note that the same calculation is used even if g_4 already contained a rating of its own: its new rating is the maximum of its original and the two discounted virtual ratings. Since the virtual ratings are discounted by an amount based on distance, only in exceptional cases will they replace an existing rating.

Spreading virtual ratings to neighbouring buckets does not, of course, enlarge what is being recommended. Recommendations continue to be made at the level of individual buckets.

3.3 Rating normalisation

Phan et al. normalize the frequency-based ratings to the range $(0, 1]$ [13]. They do this by dividing the frequency (the number of photos taken by a user in a bin) by the total number of photos taken by that user. However, we found that by their approach 98% of the normalised ratings lie between 0 and 0.1. This has several problems: it implies low preference for 98% of all locations in which a user took a photo; it gives a very skewed distribution; and it means that a recommender that is evaluated using RMSE can do well by always predicting a number between 0 and 0.1.

Instead, we follow Celma's method [4] to convert implicit feedback to a 1-5 rating scale. Following Celma, we compute the Complementary Cumulative Distribution of the frequencies in a user's profile. Then, items (buckets) that fall in the top $80 - 100\%$ of the distribution are given a rating of 5, items that fall into the $60 - 80\%$ range are given a rating of 4, and so on. Celma proposed his method in the context of music listening: to convert how often a user listens to a track into a 5-point scale. Unlike the kinds of 5-point *explicit* rating scales used in book and movie recommenders on the web, for Celma's normalized ratings, a rating of 1 does not necessarily mean that the user *dislikes* the item; rather, the fact that the item was listened to at all implies some level of *positive* feedback, but less enthusiastic positive feedback than that associated with higher points on the scale. It seems appropriate to use Celma's method for the implicit frequency-based ratings that we have in our photo-taking scenario.

[1]We use the geopy Python library for this purpose: https://pypi.python.org/pypi/geopy

Table 1: Recommender configurations

	$(0, 1]$ ratings	1-5 ratings
No virtual ratings	1	5
Virtual ratings	1-VR	5-VR

3.4 Recommendation algorithm

At this point, we have a normalized ratings matrix. Our goal, given a user and bucket for which the user has no rating, is to predict the user's rating. For this, we use matrix factorization to transform users and buckets into the same latent factor space. We choose to use matrix factorization since it is widely used for collaborative recommenders and a form of matrix factorization was the best performing approach in [13]. Specifically, for the matrix factorization we use Koren et al.'s SVD [9], solving the objective function using stochastic gradient descent. We have not 'swapped in' different recommender algorithms since our focus is on measuring the contributions made by the virtual ratings and the different forms of normalisation. (We do, however, compare against three baseline recommender systems – see below.)

4. EXPERIMENTS

4.1 Datasets

We collected the data used in this work from the photo-sharing website flickr.com by using its API. We searched for geotagged photos taken in London and Dublin in 2015 to create two datasets. After geohashing, we discarded users who had ratings for fewer than five buckets. The final dataset for London contains 112,671 photos taken by 978 unique users. Users have an average of 115 photos. The final dataset for Dublin contains 54,082 photos taken by 1,567 users and users have an average of 34 photos.

4.2 Recommenders

We compared four configurations of the recommender, depending on whether ratings were normalized to $(0, 1]$ (as in [13]) or to 1-5 (as we propose) and depending on whether virtual ratings are used or not. The names of these four configurations are given in Table 1. It follows that the system called '1' is closest to the one described in [13]: ratings are normalised to $(0, 1]$ and virtual ratings are not used. The main difference with [13], as mentioned in Section 3.1, is that we are using geohashing where they used their CASH method.

We also compare against three baseline recommenders, POP_H, POP_ALL, and HOME, which we now describe.

4.2.1 Popularity-based recommenders

We compare our methodology with a baseline recommender that recommends the most popular items for which the user has no rating. It is well-known that, in recommender systems in general, recommending the most popular items can be a highly competitive baseline [18]. For the kinds of 5-point explicit rating scales used in book and movie recommenders on the web, a popularity-based recommender typically recommends to a user those items that she has not rated and that have the greatest number of high ratings (4 or 5 stars) [18]. In other words, the recommender only recommends items that lots of people like. Accordingly, we include in our experiments a baseline that we refer to as POP_H ('H' for 'high'): in computing popularity, it counts only those users who have given the item a rating of 4 or 5. But, as we have already mentioned in Section 3.3, on the rating scales that we are using, low ratings (e.g. 1 and 2) are not signals of dislike: they show a track was listened to (in the case of music) or a photo was taken (in our case), which is positive feedback, albeit not as positive as a rating of 4 or 5. Hence, we also include another baseline recommender, POP_ALL. In computing popularity, it recommends items rated by the greatest number of users, irrespective of the values of the ratings.

4.2.2 Home location-based recommender

Using a Flickr dataset, Van Laere et al. conclude that a user is more likely to take pictures in locations that are closer to home [19]. It follows then that we can build a baseline recommender that recommends to a user locations for which she has no rating and that are close to her home location.

Some Flickr users provide a textual description of their home location in their Flickr profile. For the users in our datasets, we wrote a crawler to visit their profiles and obtain their home location descriptions, if given. (Of course, not all of these descriptions will be correct, and this may reduce the performance of this baseline recommender.) We convert the textual description to a geolocation (latitude and longitude).[2] For the London dataset, we were able to obtain the home geolocations of 40.52% of the users; for the Dublin dataset, it was 39.5% of the users.

The baseline recommender, which we refer to as HOME, works as follows. For each user whose home location coordinates are known, we calculate the geodesic distance between their home location and the centres of the buckets (items). Then we recommend the closest buckets for which the user has no rating. For users whose home location coordinates are not known, we default to recommending the most popular buckets in the dataset, as POP_ALL would do.

4.3 Methodology and metrics

Phan et al. calculate the RMSE between the predicted and actual ratings for the members of a test set. We chose to base our experiment on a newer methodology, emphasising recommendation over prediction [6]. (In any case, because two of our systems normalise to $(0, 1]$ and two normalise to a discrete 1-5 scale, we cannot directly compare their RMSEs.)

We used 5-fold-cross-validation using 80% for training and 20% for a probe set. From the probe set, we construct a test set. The idea is that the test set will contain items from the probe set that the user liked, and therefore these are ones for which a recommender will be rewarded if it recommends them in the experiments. In [6], where they assume a conventional 1-5 scale, the test set contains only the highly-rated items from the probe set (i.e. ones rated 4 or 5). However, we return to the point that we have made before that, for Celma-style normalisation of frequency data, a rating of 1 or 2 does not necessarily denote dislike. If we want to test recommender performance on all liked items, then we must include in the test set all probe items, irrespective of their values of their ratings.

In fact, we have chosen to run all experiments twice: in one set of experiments, from the probe set we create a test set by retaining only test items where the user's normalised rating is 4 or 5; for the other set of experiments, we take all of the probe set as test set.

There are no virtual ratings in the test sets and, where different recommenders are compared (even ones that normalise to $(0, 1]$), they are compared using the same sets of test items.

For each test item, we randomly select 1000 other buckets for which the user has no rating. We predict the user's ratings for all 1001 buckets and then sort them by descending predicted rating. We then recommend the top-k ($k = 10$), which may or may not include the test item.

This means that experiments are being done on an item by item basis for each item in the test set, rather than on a user by user basis. We must define our metrics accordingly.

For each test item, we measure whether the test item is in the top-k or not. If it is, we call this a hit and record the total number of hits, H. From this, we calculate the *hit rate* (or recall):

$$\mathrm{HR} = \frac{H}{|\mathit{Test}|} \tag{1}$$

where *Test* is the test set.

We also calculate the *average reciprocal hit-rank* (ARHR), which in our setting we define as follows:

$$\mathrm{ARHR} = \frac{1}{|\mathit{Test}|} \sum_{\substack{r_{ui} \in \mathit{Test}, \\ \mathit{rank}_{r_{ui}} \neq 0}} \frac{1}{\mathit{rank}_{r_{ui}}} \tag{2}$$

where $\mathit{rank}_{r_{ui}}$ is the position of this test item in the top-k ($1 \leq \mathit{rank}_{r_{ui}} \leq k$) or zero if this test item was not recommended in the top-k.

Although ARHR considers the positions of the hits, neither ARHR nor HR gives information about the original rating as well as the position. It is common to discount this based on the logarithm of the rank. Hence, inspired by discounted cumulative gain (e.g. [8]), we define *average discounted gain* (since, with only one test item at a time, it is not cumulative), as follows:

$$\mathrm{ADG} = \frac{1}{|\mathit{Test}|} \sum_{\substack{r_{ui} \in \mathit{Test}, \\ \mathit{rank}_{r_{ui}} \neq 0}} \begin{array}{ccc} r_{ui} & \mathit{if} & \mathit{rank}_{r_{ui}} = 1 \\ \frac{r_{ui}}{\log_2(\mathit{rank}_{r_{ui}})} & \mathit{if} & \mathit{rank}_{r_{ui}} > 1 \end{array} \tag{3}$$

4.4 Parameter Δ

When using virtual ratings, there is the parameter Δ, which needs to be set for the experiments. With small values of Δ, there are fewer virtual ratings than with larger values of Δ. We tried values of 0 (which is the same as no virtual ratings), 75, 150, 225 and 300. In Figures 2 and 3,

[2] We do this by using the geopy python library, which uses the Google Maps V3 geocoder: https://developers.google.com/maps/documentation/geocoding/intro

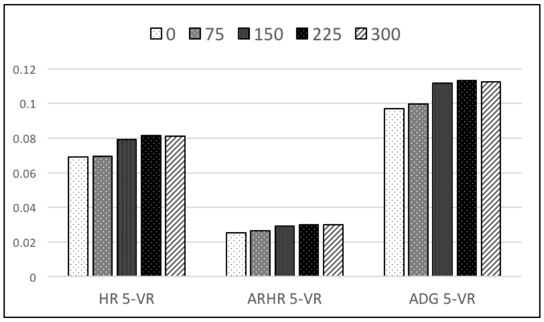


Figure 2: HR**,** ARHR **and** ADG **for 5-VR with varying** Δ**, London dataset**

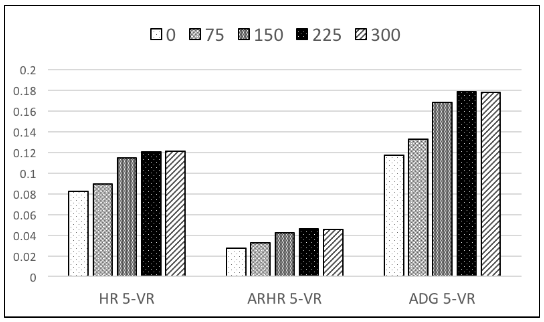


Figure 3: HR**,** ARHR **and** ADG **for 5-VR with varying** Δ**, Dublin dataset**

we show the hit rate, average reciprocal hit-rank and average discounted gain for one of the system configurations (5-VR) with varying values of Δ for the London and Dublin datasets, respectively.

As can be seen, HR, ARHR and ADG tend to increase as Δ increases but then, as Δ becomes too big and so ratings are being spread too far, HR, ARHR and ADG level off or even fall. The Figures show that the most competitive value for Δ for this system configuration for both datasets is 225.

The results for 1-VR (not shown) follow a similar pattern, but its most competitive value for Δ is 300 for both datasets.

These are the values we use for Δ in the results that we show in the next section.

4.5 Results

We now compare the seven recommenders, i.e. the three baselines (POP_H, POP_ALL and HOME) and the four configurations of our recommender, which depend on the normalisation scheme and whether virtual ratings are used or not (1, 1-VR, 5 and 5-VR). As per the previous section, the two configurations that use virtual ratings use their most competitive values for Δ.

Figures 4, 5 and 6 show the hit rate, average reciprocal hit-rank and average discounted gain respectively.

The worst-performing system overall is the one designated 1, corresponding roughly to the system in [13]. Not only is it out-performed by the other three configurations (1-VR, 5 and 5-VR), it is out-performed by the three baselines (except on the London dataset, where it has very slightly higher HR than HOME does, but even here it is out-performed by the other two baselines).

The best-performing system is 5-VR — the one that combines both our innovations. It out-performs all recommenders on all metrics across both datasets.

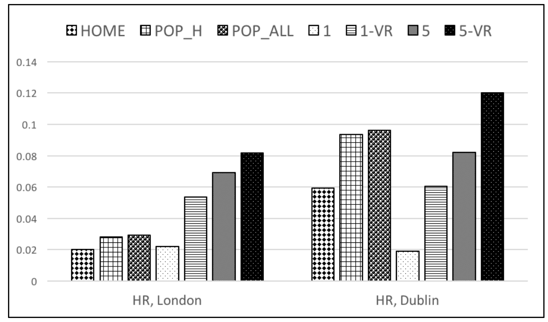


Figure 4: HR

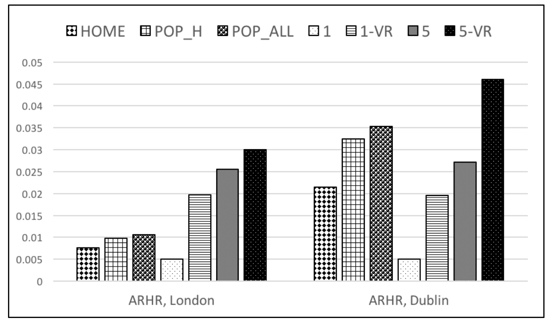


Figure 5: ARHR

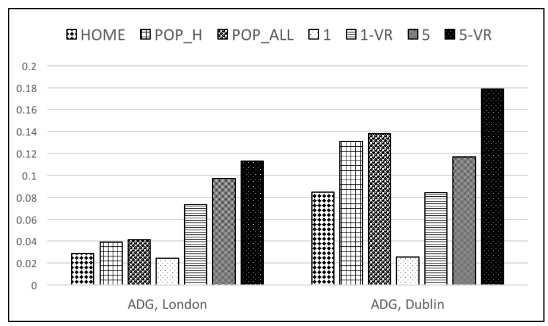


Figure 6: ADG

There are also some other patterns in the results. For both datasets and all evaluation metrics, of the four non-baselines, system 1 is the worst; adding virtual ratings (1-VR) makes an improvement; scaling to 1-5 (system 5) is better again. But the best-performing configuration is 5-VR, where we scale to 1-5 and use virtual ratings.

Another pattern is that the baselines are more competitive on the Dublin dataset than the London dataset. They never out-perform system 5-VR, but on the Dublin dataset, POP_H and POP_ALL both out-perform systems 1, 1-VR and 5. The performance of the HOME baseline is more mixed.

For reference, Table 2 shows the values on which the Figures in this section are based.

In fact, as explained in Section 4.3, we chose to run two set of experiments, differing in the way in which the test set was constructed from the probe set. In one case, all ratings

Table 2: Results for experiments where test set equals probe set

Dataset	Metric	HOME	POP_H	POP_ALL	1	1-VR	5	5-VR
London	HR	0.02	0.0282	0.0294	0.0222	0.0536	0.0691	0.0816
London	ARHR	0.0077	0.0098	0.0106	0.005	0.0197	0.0255	0.0301
London	ADG	0.0287	0.0394	0.0411	0.0243	0.0732	0.0971	0.1131
Dublin	HR	0.0595	0.0936	0.0963	0.0191	0.0603	0.0821	0.1203
Dublin	ARHR	0.0215	0.0324	0.0354	0.005	0.0195	0.0272	0.0461
Dublin	ADG	0.085	0.1308	0.138	0.0255	0.0842	0.1169	0.1788

from the probe set are placed into the test set. These are the experiments whose results were shown in the Figures and in Table 2. In the other case, only highly-rated items from the probe set are included in the test set. The results for this set of experiments are given in Table 3. Comparing the two tables, we see that numerically the results in the second set of experiments are higher. But, for the most part, the story about which systems out-perform each other remains the same. In particular, 5-VR remains far and away the best of the recommenders.

5. DISCUSSION & CONCLUSIONS

We presented an approach to recommending geolocations to users for photo-taking. We geohashed coordinates to cells in a rectangular grid, and used these as the items in an implicit feedback matrix. We investigated two innovations. One was to create virtual ratings in neighbouring cells. The other was to normalise ratings using a method developed for music recommenders. Our experiments, measuring hit rate, average reciprocal hit-rank and average discounted gain, showed that the two innovations together outperformed all other configurations and popularity-based and home location-based baseline recommenders.

There are many avenues for future work. For a start, we can test on other datasets and on recommender algorithms other than SVD. There remains an open question about the precision of the geohashing. Here we are using precision of 7, resulting in buckets that are about 150 metres by 150 metres. The only way to determine whether this is the best choice or whether higher precision (smaller buckets) would be better is through a user trial. It may even be that precision should be personalised or adaptive in some other way, and this could be investigated in future work.

Finally, there is the opportunity to integrate other factors into the work. Presently, we recommend only the photo-taking location. Date and time may also be important: perhaps a location may offer better candidate subjects for photos in certain seasons, on certain dates or at certain times of day. Related to date and time are the photo-taking conditions: a bucket may be a better location, e.g., when the sun has risen, when the weather is not overcast or when the sun is not glaring (although, of course, for some photographers these difficult conditions provide opportunities for the exercise of their photographic talents). Some users might also want assistance with the choice of subject, camera settings or composition. For this, it may be possible to integrate our work with the kind of work done on the social camera [2] and ClickSmart systems [15].

6. ACKNOWLEDGEMENTS

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

7. REFERENCES

[1] P. Bhargava, T. Phan, J. Zhou, and J. Lee. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *Procs. of the 24th Int. Conf. on the World Wide Web*, pages 130–140, 2015.

[2] S. Bourke, K. McCarthy, and B. Smyth. The social camera: Recommending photo composition using contextual features. In *Procs. of the Second Workshop on Context-Aware Recommender Systems*, 2010.

[3] X. Cao, G. Cong, and C. S. Jensen. Mining Significant Semantic Locations from GPS Data. *Proc. VLDB Endow.*, 3(1-2):1009–1020, 2010.

[4] Ò. Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 2010.

[5] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *Procs. of the 18th Int. Conf. on the World Wide Web*, pages 761–770, 2009.

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Procs. of the 4th ACM Conf. on Recommender Systems*, pages 39–46, 2010.

[7] B. Hidasi and D. Tikk. Approximate modeling of continuous context in factorization algorithms. In *Procs. of the 4th Workshop on Context-Awareness in Retrieval and Recommendation*, pages 3–9, 2014.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[9] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[10] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. Lars: A location-aware recommender system. In *Procs. of the IEEE 28th Int. Conf. on Data Engineering*, pages 450–461, 2012.

[11] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning Geographical Preferences for Point-of-interest Recommendation. In *Procs. of the 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 1043–1051, 2013.

[12] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In *Procs. of the 30th AAAI Conf. on Artificial Intelligence*, 2016.

Table 3: Results for experiments where test set contains highly-rated probe items

Dataset	Metric	HOME	POP_H	POP_ALL	1	1-VR	5	5-VR
London	HR	0.0272	0.0413	0.0402	0.0315	0.0698	0.0932	0.1083
London	ARHR	0.0104	0.0137	0.0146	0.0071	0.0261	0.0361	0.0416
London	ADG	0.0657	0.0923	0.0938	0.0577	0.165	0.2258	0.26
Dublin	HR	0.0759	0.1193	0.1241	0.0289	0.081	0.1139	0.1529
Dublin	ARHR	0.0292	0.0442	0.0479	0.0084	0.028	0.0403	0.0629
Dublin	ADG	0.1812	0.2809	0.2962	0.0618	0.185	0.2638	0.3828

[13] T. Phan, J. Zhou, S. Chang, J. Hu, and J. Lee. Collaborative Recommendation of Photo-Taking Geolocations. In *Procs. of the 3rd ACM Multimedia Workshop on Geotagging and its Applications in Multimedia*, pages 11–16, 2014.

[14] D. Quercia, R. Schifanella, and L. M. Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Procs. of the 25th ACM Conf. on Hypertext and Social Media*, pages 116–125, 2014.

[15] Y. S. Rawat. Real-time assistance in multimedia capture using social media. In *Procs. of the 23rd ACM International Conference on Multimedia*, pages 641–644, 2015.

[16] Y. Shi, P. Serdyukov, A. Hanjalic, and M. Larson. Nontrivial Landmark Recommendation Using Geotagged Photos. *ACM Trans. Intell. Syst. Technol.*, 4(3):47:1–47:27, 2013.

[17] B. Sigurbjörnsson and R. van Zwol. Flickr Tag Recommendation Based on Collective Knowledge. In *Procs. of the 17th Int. Conf. on the World Wide Web*, pages 327–336, 2008.

[18] H. Steck. Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 125–132. ACM, 2011.

[19] O. Van Laere, S. Schockaert, and B. Dhoedt. Georeferencing Flickr resources based on textual meta-data. *Information Sciences*, 238:52–74, 2013.

[20] Y. Yang, Z. Gong, and L. H. U. Identifying Points of Interest by Self-tuning Clustering. In *Procs. of the 34th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 883–892, 2011.

[21] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *Procs. of the 36th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 363–372, 2013.

[22] N. Zheng, Q. Li, S. Liao, and L. Zhang. Which photo groups should I choose? A comparative study of recommendation algorithms in Flickr. *Journal of Information Science*, 36(6):733–750, 2010.

Learning the Popularity of Items for Mobile Tourist Guides

Patrick Hiesel
Technical University of Munich
Boltzmannstraße 3
85748 Garching, Germany
hiesel@in.tum.de

Matthias Braunhofer
Free University of
Bozen-Bolzano
Piazza Domenicani 3
39100 Bolzano, Italy
mbraunhofer@unibz.it

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

ABSTRACT

A context-aware recommender system incorporates the knowledge of different contextual factors such as time or weather information to improve item suggestions made to a user. This requires the system to have a large knowledge base for inferring contextual information and enabling accurate and timely recommendations. We present a versatile approach for a context-aware recommender system in the tourism domain by crawling publicly available information from a variety of sources and learning the contextual popularity of points of interest based on a generalized check-in model. We have deployed a test instance of our system for the greater area of Munich and the German state of Bavaria. Analyzing the results from the offline learning has led to interesting insights including when and in which weather conditions certain items are popular.

CCS Concepts

•Information systems → Recommender systems;

Keywords

Recommender systems; context; data analytics; mobile guides

1. INTRODUCTION

Recommender systems are a composition of software tools and techniques that suggest items to users that are likely to be interesting to them and relevant to their needs [10]. Traditional recommender systems consider items liked/rated by users in the past and possibly some additional information such as item characteristics to estimate the ratings for items that the users have not yet consumed [1]. Applications range from suggesting products that have been bought together by other users in the past over suggesting people a user might know based on their existing list of friends to suggesting music based on genres listened to.

Context-aware recommenders enhance traditional recommender systems by incorporating the knowledge of different contextual factors - such as time or weather information - to further improve the item suggestions made to a user [1]. These systems seek to better match users and their current context with items that are popular in the same or similar contexts.

One major application area of context-aware recommender systems is travel and tourism, where scenarios are significantly more complicated than traditional user-product matchings [3]. In addition to the general preferences of a user, the utility and relevance that a point of interest (POI) has to a user heavily depends on the user's current context. A beer garden, for instance, would provide a higher value to the user on sunny summer days rather than on rainy winter days and a car that knows a driver's route, fuel level and gas prices can make better suggestions for gas stations to refuel. This is especially important in the scenario of a proactive recommender system [14], i.e., a recommender that pushes item suggestions to the user based on the current situation (e.g. location, time of day or weather) without explicit user request.

In our research, we propose, implement and evaluate a novel approach for a context-aware recommender system in the tourism domain by aggregating publicly available information from a variety of sources and learning the contextual popularity of POIs based on a generalized check-in model. We aggregate different types of data, including POIs, check-ins and contextual information to build a knowledge base and infer knowledge about the contextual popularity of items focussing on aspects of temporal and geographic context. The gained knowledge can also be utilized to mitigate cold-start problems when no or little information about the user is available.

In the following, we first explore related work (Section 2), describe the context model, data sources and system design of our implementation (Section 3) and then present the process and results of analyzing the data (Section 4). Finally, Section 5 draws conclusions and discusses open future work directions.

2. RELATED WORK

Context-aware recommender systems have been a topic of growing research interest in the recent years and aim at generating more relevant recommendations by adapting to the specific contextual situations of the user and the recommended items (e.g., weather, temperature, season and mood) [1]. There exist numerous commercial and research systems, such as Foursquare, Yelp, South Tyrol Suggests (STS) [4] and ReRex [2], that have already been successfully implemented and that exploit the current user's and item's context when recommending items. These systems

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

use different approaches to incorporate context into the recommendation process. Roughly, these approaches can be divided into three categories [1]: (i) contextual pre-filtering, where context is used for selecting the relevant set of ratings before computing predictions with a traditional, two-dimensional prediction model; (ii) contextual post-filtering, where context is used to adjust the recommendation list resulting from a two-dimensional rating prediction model; and (iii) contextual modeling, where context is directly incorporated into the prediction model.

Most current context-aware recommender systems work in pull mode, i.e., the user has to explicitly make a request (pull) for recommendations, possibly by entering information about her preferences, needs and constraints. A new generation of context-aware recommender systems, called proactive recommender systems, are instead pushing recommendations to users without their specific request, when they are in a contextual situation that the system considers as suitable for the recommendations [14]. Despite the advantages of proactive recommender systems - especially in mobile usage scenarios - relatively little research has been conducted specifically on this topic. One example is [14], where the authors proposed a proactive recommender system model consisting of two phases: (i) the situation assessment phase, which evaluates whether or not the current contextual situation calls for a recommendation; and (ii) item assessment phase, which is only executed when the first phase indicates a promising situation and assesses the candidate items to finally decide which items should be pushed to the user as recommendations. Subsequent work in [13] has evaluated the effectiveness of the proposed model by applying it to a restaurant recommender system, and found that users highly appreciate proactive recommendations if they are relevant and properly timed.

In another work, Dali et al. [5] presented different timing models based on random forest to classify user contexts that are suitable for recommendations and user contexts in which users are highly likely to refuse any recommendations. Results from a user study revealed that a hybrid model that first decides whether it should use a personal or non-personal timing model, and then classifies whether the context is suitable for recommendations is superior to both the personal or non-personal timing models.

In another study, Pielot et al. [9] showed that boredom can be inferred from patterns of mobile phone usage and that users are more likely to appreciate proactive recommendations during inferred phases of boredom. Hence, they concluded that using boredom as trigger independent from content might help to make proactive recommendations a more pleasant experience for users.

Finally, Borrís et al. [3] survey intelligent recommender systems in travel and tourism and also mention context and proactivity as important factors.

3. CONTEXT, DATA SOURCES AND SYSTEM DESIGN

In order to create a versatile system that can gather relevant data for any geographic area and infer the contextual popularities of POIs, we first discuss the context model that our work is based on. This leads to the features we need to extract from the data we crawl. We then discuss different data sources for items, check-ins and context and present a brief system design of both the backend implementation and a corresponding mobile client.

3.1 Context Model

Our research is based on a context model for proactivity in mobile recommender systems defined by Wörndl et.al. [14]. The model relies on domain-dependent context modeling in four distinct categories: *user context, temporal context, geographic context* and *social context*. The recommendation process itself is divided into two phases to first analyze the current situation and then examine the suitability of particular items. This allows for both a proactive and passive recommender system.

The context model defines the data we need to aggregate and the features that should be extracted. In particular, we want to infer the temporal and geographic context of items. The user context is extracted directly in our mobile app prototype in a later stage. The social context is neglected in our prototype but could be integrated in our approach. Based on this context model, we focus on the following contextual features: *weather condition, temperature, season, day of week, time of day* and *time of year*.

3.2 Data Sources

The overall goal of this work to recommend POIs in a mobile tourist guide based on publicly available data sources. We therefore need to acquire and aggregate data in three categories: items (the POIs), check-ins (to determine the popularity of items) and context.

3.2.1 Items

The POIs form the foundation of our system as the user's overall perception of the system first and foremost depends on the quality and suitability of the recommended items. We therefore designed a general model for an item in our domain that can then be populated with data. Core data fields include: name, description, location, images, phone, rating, street, city, country. While the core fields should always be populated, each data source's crawler can define additional fields it wants to persists. Based on this model, we examined different, publicly accessible data sources: Foursquare, Yelp, Quermania, Facebook, Wikipedia, Open Street Maps. We designed and implemented crawlers for each of those sources and aggregated over 175,000 items for the German State of Bavaria.

3.2.2 Check-ins

Based on this knowledge base of 175,000 items, we want to infer the popularity in accordance to the context model. Our premise is that a place is popular if many people visit it. It follows that a place is popular in a certain context (e.g. on a sunny Saturday evening in summer) if many people visit it when this context condition applies.

To infer this knowledge, we must first define a model for determining how popular a place is at a given time or any other context. We base this model on a generalized check-in. In our model, a check-in can be any evidence that a user visited a place at a certain time. This includes an explicit check-in on Foursquare or Facebook, as well as implicit check-ins by taking a picture or generating a GPS trace on a smartphone. A check-in marks a singular point in time. To make an educated judgement on how many people are present at any point in time, we must further make an assumption about the average duration of a visit. There has been research on the activity duration of different activities [8] that we use to infer the number of present users from check-ins at any point in time based on the POI type.

We looked at different publicly available data source to crawl check-ins from including Flickr, Twitter and Foursquare. Flickr is a photo sharing platform with a rich set of geo-tagged photos. A geo-tagged photo contains the latitude and longitude of the place it was taken as well as a level of accuracy of this information. In addition, a textual description is provided for most of the images. We used Flickr as the primary source of check-ins for rural areas, establishing a place-to-check-in mapping in a sparsely populated area which is more straightforward given the geo coordinates of both places. In our research, we have sampled about 249,000 images to be used as check-ins.
Twitter offers a large number of tweets that can be associated with a place and hence counted as a check-in. There are two main features, that can be used for association: geo-tagged tweets and Twitter Places. Geo-tagged tweets carry a latitude and longitude and can be associated to places using geo-fencing. While this is a valid approach in rural areas, it is insufficient for cities due to the number of places being close to each other. Twitter places specify a specific geographic place, such as a city or a restaurant and can be mapped to a POI directly. In our research, we have sampled 29m tweets in about eight weeks using Twitter's Streaming API.
Foursquare has an explicit check-in feature - similar to Facebook - that would let users check-in at a place using a button in their app. In addition, Foursquare samples the user's location on the smartphone to proactively recommend POIs. While this data seems promising, none of it is publicly accessible and was therefore not used for our prototype. On the contrast, some users link their Twitter account with their Foursquare account resulting in each explicit check-in triggering a public tweet. Following the approach proposed by Melia and Segui [8], we extracted this information from the tweets we collected and linked them to Foursquare venues in our database. Based on our Twitter dataset of 29m tweets, we successfully linked 2.9m tweets to a Foursquare Swarm POI. The association rate for the German state of Bavaria was 8 check-ins (tweets) per 1,000 POIs and enriched our dataset for cities.

3.2.3 Context

Based on the aggregated knowledge base of items and check-ins we now want to add a third data source to enrich our data with context. We hereby focus primarily on dimensions of the geographic and temporal context. Given the timestamp of a single check-in we can infer all attributes of the temporal context using a static calendar library. We therefore use java.util.Calendar to infer *season, day of week, time of day* and *time of year*. To infer the weather condition and temperature of check-ins we added Wunderground's Weather History API that can be used to provide the weather for a place at any given time in the past. Using these two sources, we were able to infer the temporal and geographic context for all check-ins.

3.3 System Design and Server Implementation

We implemented our prototype in Scala using the Play! framework. Scala is a language that is executed on top of the Java Virtual Machine and is fully interoperable with Java. The language combines object-oriented programming with functional programming which makes operations on arrays and lists easy to implement. On top, all Java libraries are usable in Scala which is a major benefit. The Play! framework offers a modern web MVC architecture with a simple control flow and the advantage of using templates in the views. We have combined this with Slick - a functional relational mapper - to have a persistent database layer based on a performance-tuned MySQL instance. Slick offers a clean way of accessing and filtering persistent data sets using Scala's functional API.

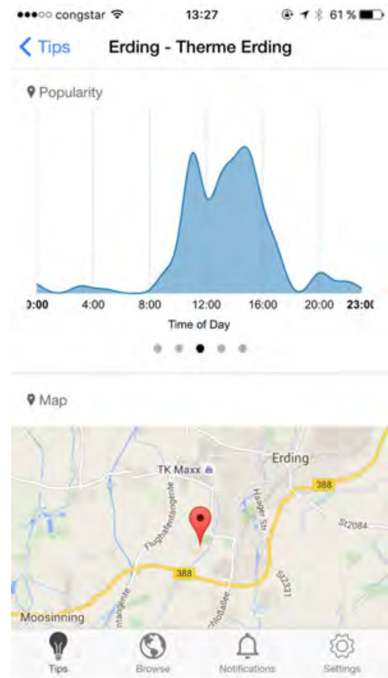


Figure 1: Screenshot of the Mobile App with the Popularity Graph for Time of Day

Data aggregation is either done through a REST API (if available) or through one of our crawlers. In the course of this research, we have created multiple crawlers and spiders on top of JSoup to extract knowledge from publicly available sources. We used Amazon AWS resources to carry out parts of these tasks with on-demand resources, while maintaining only one dedicated server at all times. We used SQS, a distributed pipe service, to communicate between the different servers.

3.4 Mobile Application Design

We have also designed, developed and tested a user interface concept to investigate how to communicate contextual information about recommended items to the user in a mobile tourist guide [7]. Thereby, the user can retrieve information about interesting POIs and review various graphs about the item popularity in an item detail screen. We show a textual summary of the popularity peak for both weather and day/time. We also render line and bar charts to show the popularity across different context dimensions, such as day of the week, time of day (see Figure 1 for an example) and season. The mobile application was implemented using the cross-platform framework Ionic and is fully functional.

To receive early feedback for our concept, we evaluated our system in a user study with 14 subjects. The participants were asked to test the application for two weeks and then complete a survey about their experience with the user interface elements. The study results indicated that the popularity inference and graphs provide benefits for users. Users stated that popularity graphs assisted them while deciding which place to visit. More detailed results of this preliminary study can be found in [7].

The focus in this paper is not on the user interface but on how to analyze the collected data and infer insights that can then be integrated in a mobile tourist guide. We present our course of action and the gained results of the offline learning in the next section.

4. OFFLINE ANALYSIS AND PROBABILITY LEARNING

To infer basic popularities for different conditions we follow the data analytics process proposed by Runkler [11]. The process decomposes the data analytics pipeline into four steps: preparation, preprocessing, analysis and post-processing. Following this process, we have identified the following tasks for each step:

1. Preparation: identifying goals and research question; data collection.
2. Pre-processing: merging POIs from different sources; associating POIs with check-ins filtering, sampling and discretization, normalization.
3. Analysis: visualization, popularity inference and prediction.
4. Post-processing: evaluation.

To follow this process, we define an overall research question and subquestions that we seek to answer.

- Overall research question: how can we learn (infer) the popularity of POIs for a context-aware recommender system?
- Subquestion 1: how can POIs from different sources be matched and duplicates be eliminated?
- Subquestion 2: how can POIs be associated with check-ins?
- Subquestion 3: given a set of check-ins for a POI, how can we infer the popularity under different temporal conditions?
- Subquestion 4: given a set of check-ins for a POI, how can we infer the popularity under different geographic conditions?
- Subquestion 5: given a set of check-ins for a POI and the base popularities, which algorithms are suitable for making a compound decision/recommendation?

In the following, we present our approaches and results on these tasks according to the subquestions we are trying to answer.

4.1 Merging POIs from Different Sources

We have added different POI sources to our system, including Foursquare, Yelp and Quermania. Especially Foursquare and Yelp provide a lot of overlapping data, since both have restaurants and bars in their database. This leaves us with the problem of identifying and merging co-referent POIs, as we do not want to show or recommend the same POI multiple times.

Merging co-referent POIs has been extensively studied and there are multiple approaches to the problem, e.g., using a fuzzy set and probability theory [12] or a DBSCAN, a common clustering algorithm [6]. The fuzzy set approach assumes, that the majority of POIs are user-generated and therefore has large differences in between two versions of a POI's name. Our data differs from the assumptions of these papers in a way, that it is already pre-filtered by Yelp and Foursquare. We therefore take a simple approach matching POIs from different sources: we compare the Levenshtein distance of the names of the two POIs in question and investigate their geographic distance.

4.2 Associating POIs with Check-Ins

Associating POIs with check-ins is a non-trivial task, given that the data comes from entirely different sources. Depending on the source of the POI and the check-in, we have identified several ways of creating an association:

4.2.1 POIs with Flickr/Twitter Check-ins

Flickr and Twitter both provide data records that have coordinates, a timestamp and some textual data like the tweet's content or the Flickr image's caption. Associating these records with a POI presents a hard problem that can be tackled in many different ways:

Simple Geofencing: The computationally easiest option is to use a rectangular geofence around a POI and associate each check-in record within this area with the POI. This works well for exposed POIs (i.e., satellite POIs that have no other POIs around them), but is inapplicable to the majority of POIs in cities including restaurants and bars.

Advanced Geofencing: To improve both the FPR (false positive rate) and FNR (false negative rate) of the simple geofencing approach, we propose a more complex way of setting up the geofence. Depending on the POI's type, a more complex geofence structure can range from a smaller circle (suitable, e.g., for bars or restaurants in cities) to a polygon following the shape of ski slopes or trails.

Clustering: can be done in a supervised, unsupervised or semi-supervised fashion. A supervised clustering approach would assume that we have a couple of check-ins per POI where we have obtained evidence that they belong to this POI. Other nearby check-ins could be associated using algorithms like KNN (see Evidence-based Clustering below). An unsupervised approach could use algorithms like DBSCAN to discover clusters in unclassified data. Once the algorithm has discovered clusters, we could use different techniques to associate POIs with clusters.

Evidence-based Clustering: As mentioned with simple clustering, one approach could be to use supervised or semi-supervised clustering for association. To start this algorithm we would need a base dataset of check-ins that are associated with POIs. One approach to generate such a dataset would be to use evidence from the metadata. Most tweets/Flickr descriptions contain the topic of the tweet or image such as "Neuschwanstein". Using simple word-matching or more complex NLP techniques we would obtain a base set of check-ins to use for clustering and association.

We use Flickr and Twitter check-ins for exposed sights to deliver a proof of concept for our pipeline and model and could therefore use simple geofencing to associate check-ins with POIs (by a square whose size depends on the POI type). Our approach has linked 280,983 check-ins with 177 sights. While this approach was sufficient for our use case, future work could lie in exploring other association techniques to make using these check-in sources viable for more densely populated areas.

4.2.2 *Foursquare POIs to Check-ins using Twitter*

As briefly outlined before, Melia and Segui [8] proposed an approach to aggregate public Foursquare (Swarm) check-ins using Twitter. Foursquare users that have linked their account with Twitter will automatically publish a tweet if they check-in publicly. By analyzing the corresponding tweets using our data processing pipeline, we obtain the unique Foursquare ID of the POI where the user checked-in. In addition, we obtain the timestamp of the tweet and hence the check-in.

This approach is promising, as it established a definitive relation between a check-in and a POI (i.e., there are no false-positives). We have aggregated a mixed data set of tweets containing both geo-located tweets and tweets linked to public Foursquare and Swarm check-ins using the streaming API over eight weeks. The dataset has 29m tweets in total, 2,9m of which we have successfully linked to a Foursquare or Swarm POI.

Our prototype only incorporates POIs in the German State of Bavaria. We could therefore only use a small subset of the 2.9m tweets that link to a POI in our region. We could match tweets and POIs at a rate of 8 check-ins (tweets) per 1,000 POIs. This number is relatively low compared to the effort it took to obtain the data using the pipeline we outlined earlier. While this was not an ideal fit four our use case, the approach can be a better choice in regions with higher Foursquare adoption (such as New York or San Francisco).

4.3 Filtering, Sampling and Normalization

Until this point, we have crawled POIs, check-ins and context and associated POIs with check-ins. The next step is to filter the dataset to increase its quality, decide on sampling for context parameters and perform normalization if needed.

Filtering: For the qualitative evaluation and our mobile app prototype we focus on exposed sights out of our large POI dataset and have identified 176 sights with a high POI to check-in matching having both a low FNR and FPR. Furthermore, we filter out POIs that have less then 400 check-ins (on average each POI has 1570 check-ins) to retain a good accuracy for all contextual popularities. All contextual popularity groups partition the remaining check-ins into a maximum of 7 groups yielding 57 check-ins on average per group. This filtering gives us a set of 114 sights in the greater area of Munich and the German State of Bavaria that we use for further processing.

Sampling: The next step to aggregate the popularity is to define what dimensions should be explored and if a dimension is continuous or discrete. We hereby explore temporal and geographic dimensions separately. Discrete variables are easy to handle when it comes to inferring the popularity, as they can be represented by a fixed number of buckets (e.g., seven for *Day of Week*) and assigning check-ins to buckets is trivial. Continuous variables however, are more difficult as questions like "How popular is this POI at 10am?" can not be holistically answered using sampling and buckets, since when settling on a fixed bucket size (e.g., half an hour) the question "How popular is this POI at 10:11am?" can not be answered accurately. When decreasing the bucket-size, buckets will only have a few check-ins as a check-in only marks a specific timestamp making the popularity impossible to infer.

We therefore explored related work on activity duration [8] to assign a presence window to each check-in accounting for the time a user was present at the POI. The activity duration depends on the POIs category and ranges from 7:43 min for breakfast to 19:01 min for dinner. Parks and outdoor have a mean activity duration of 11:21 min. Given this knowledge, we set a bucket-size of 5 min and count the check-in towards three buckets for sights. In this approach we model a user's presence at a given time by adding 1 to three buckets. With this information, we can accurately answer questions in the form of "How popular is this POI at 10:11am?", as we have a bucket ranging from 10:10:00am to 10:14:59am. However, with this model we neglect the uncertainty at the beginning and end of the activity interval. The source of this uncertainty is the fact, that we only know one timestamp and infer the user presence window through (assumed) activity durations yielding a high uncertainty at the beginning and end of this interval. An alternative modeling approach could be to use Gaussians to augment presence intervals. This way, we can model the uncertainty in a statistically correct way.

We regard the Gaussian modeling as future work and base our presence interval on activity durations without augmenting Gaussians. We introduce *feltTemperature* as a discretization of the continuous temperature as this simplifies inference and makes our predictions easier to understand for users. Table 1 shows our set of inferred contextual variables for both the geographic and temporal context.

Variable	Type	Domain
dayOfWeek	D	{Mon,...,Sun}
timeOfDay	C	[0,23]
season	D	{Spring,Summer,Fall,Winter}
feltTemperature	D	{Hot,Warm,Pleasant, Mild,Cold,Freezing}
weatherCondition	D	{Sunny,Cloud,Rainy,Snowy}

Table 1: Temporal/Geographic Dimensions for Sampling. D=Discrete, C=Continuous

Normalization: Figure 2 shows an aggregation of all of our check-ins on a weekly basis using the presence window approach based on a presence window of 1 hour. One can clearly see from the graph, that about twice as much check-ins are made during weekends, than there are on a weekday. We have therefore thought about normalizing the data as a whole to have the same relative amount of check-ins per discrete bucket (e.g. Mon-Fri).

Figure 2: All Check-ins Aggregated on a Weekly View

After investigating our data in close detail, we decided against normalization as users tend to visit more POIs on

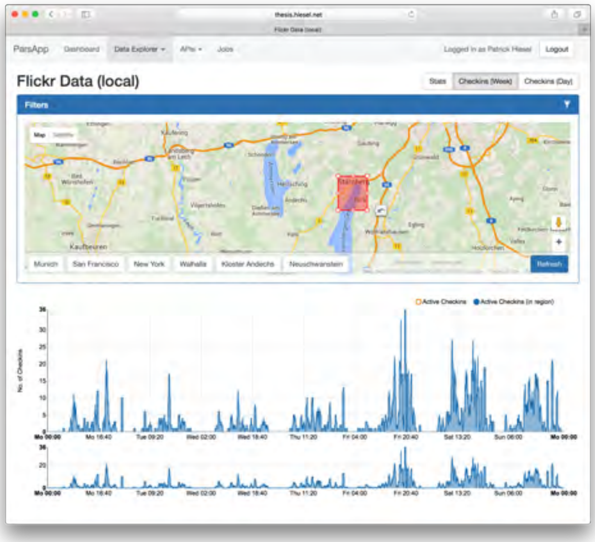


Figure 3: Geofilter Analysis

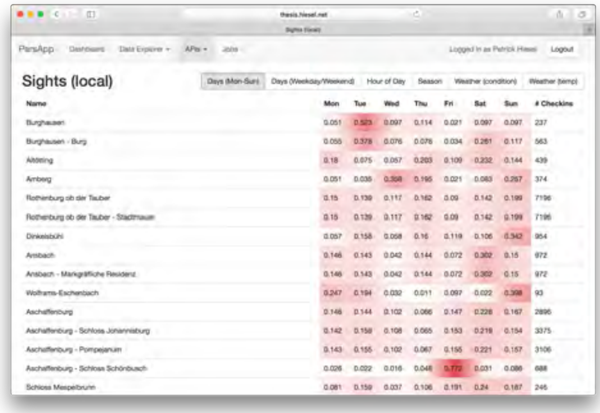


Figure 4: Popularity

the weekends/evenings when they spend leisure time. We therefore anticipate, that the weekend cliffs (as well as some other data patterns) are correct and beneficial for the sake of popularity inference.

4.4 Visualization, Popularity Inference and Prediction

4.4.1 Visualization

To effectively visualize our data, we created multiple analysis tools based on our server-side software. For effective analysis of the check-in data we build a tool to visualize different features of geo-based check-ins without an association to any POI. Our tool, as depicted in Figure 3, is able to apply a geofence to filter check-ins and compare the selected area to a baseline of all known check-ins which makes it easy to spot interesting patterns. Figure 3 shows all check-ins on a weekly level for the northern end of Starnberger See. This is a good example for the effectiveness of our tool, showing that this part of the lake is popular on Friday afternoon as well as on the weekend.

After an initial analysis using these tools, we started to model the popularity inference. For this purpose, we created different views, such as the one depicted in Figure 4, to visualize the inference outcome. Our view shows the probabilities (popularities) under different temporal and geographic conditions highlighting them with a gradient in red color such that one can easily spot patterns in the data.

4.4.2 Popularity Inference

Until this point, our analysis and visualizations yielded promising patterns. We now take a probabilistic approach to infer popularities for different contextual situations. The outcome we seek is the popularity P of a POI given a context C:

$$p(P|C) \tag{1}$$

When thinking about context and different dimensions as described before it would be most beneficial to split up C into temporal and geographic dimensions and their respective parameters (temperature t, weather condition wc, season s, weekday wd, etc.). This yields probabilities of the form:

$$p(P|t),\ p(P|wc),\ p(P|s),\ p(P|wd) \tag{2}$$

Having these base popularities would allow for compound calculations yielding the probability for a visit V to a certain POI given a situation with fixed contextual parameters:

$$p(V|C,P) \tag{3}$$

The concept of a visit is that given that a user will certainly visit a POI, what is the probability that the visit will happen at the context C.
There are different approaches to model the statistical dependence of context factors like weather condition and temperature (e.g., it does not snow when having 20°C) including Bayesian networks. While a full Bayesian model can increase accuracy, it requires a high degree of domain knowledge about the data and all parameters that is hard to obtain. We therefore use summation to infer the plain popularities for each context parameter:

$$p(P|wd = Monday, V = Walhalla) \tag{4}$$
$$= p(wd = Monday|V = Walhalla) \tag{5}$$

$$p(wd = Monday|V = Walhalla) \tag{6}$$
$$= \frac{\sum_{CheckinsForPOIOnMonday}}{\sum_{CheckinsForPOI}} \tag{7}$$

This approach yields a probabilistic distribution for each POI and each context dimension as depicted in Figure 4 which lets one judge under which conditions a place is popular. We can use prediction and the Bayes' theorem to make compound recommendations.

4.4.3 Prediction

With prediction or recommendation, we seek to answer the question "Where should I go (given the current timestamp and weather)?". With respect to our model, this would mean: what is the probability (or score) that I should choose a POI for my visit V given the inferred popularities of all

POIs P and the current context C:

$$p(V|C,P) \tag{8}$$

We propose two different approaches to answer this question: weighted additive scoring and using the Bayes' theorem.

Weighted Additive Scoring.

The first approach to obtain a popularity score given a context situation C is to use weighted additive scoring.

$$S(C,V) = \sum_{i \in [wc,t,d,s,h]} \alpha_i \; p(i,V) \tag{9}$$

For each POI we add the popularity for a specific set of context dimension (weather condition wc, temperature t, day d, season s, hour h) that we want to predict a score for. Furthermore, we multiply each dimension by a weighting factor α to make the model flexible. While we used static weights, future work could include learning of α through online of offline learning techniques.

While this model works well, it does not respect the different number of check-ins between POIs. Thus a POI with only 400 check-ins might outperform a POI with 4,000 check-ins, as the popularity values are better, while - in absolute visitor numbers - the second POI outperforms the first. This issue is addressed by an approach using Bayes' theorem.

Bayes Theorem.

Bayes' theorem can be used to inverse a dependent probability. Using summation to obtain the base probabilities as outlined in the previous section yields probabilities in the form of:

$$p(wd = Monday|V) \tag{10}$$

In other words, this means: "given that I visit a POI, what is the probability I would visit it on Monday?" (or: "what is this POI's Monday popularity"). When recommending items, we would like to answer questions of "It is Monday, which POI should I visit?":

$$p(V|wd = Monday) \tag{11}$$

So for a concrete POI (Walhalla) we can use Bayes' theorem to get the probability for a visit given our limited set of POIs and the premise that the user will visit one of these:

$$p(V = Walhalla|wd = Monday)$$
$$= \frac{p(wd = Monday|V = Walhalla)p(V = Walhalla)}{p(wd = Monday)} \tag{12}$$

All of the parameters from this model can be retrieved from our dataset:

$$p(wd = Monday|V = Walhalla) \tag{13}$$

(as obtained in the previous section)

$$p(V = Walhalla) = \frac{\sum_{CheckInsAtWalhalla}}{\sum_{AllCheckIns}} \tag{14}$$

$$p(wd = Monday) = \frac{\sum_{AllCheckInsOnMonday}}{\sum_{AllCheckIns}} \tag{15}$$

We found, that using a Bayesian approach disproportionally favors POIs that have a high number of check-ins even when being relatively unpopular and therefore used weighted additive scoring.

Using either one of those strategies leaves us with a probability (popularity) for each POI in the database, which can be used for ranking in combination with a standard weighted additive scoring approach for recommending a POI based on the different context factors.

4.5 Evaluation of Post-Processing

We discussed different types of evaluations for this part of our research with peers from the field of machine learning. While machine learning and data analytics applications are usually evaluated using cross-validation or any other sort of quantitative offline evaluation, we did not see a fit for these types of evaluations given our data and inference schema. We therefore decided to first evaluate our research by qualitatively assessing inferred popularities for selected results in a case study and discussing interesting findings and patterns. The ultimate goal is to integrate the results about item popularities into the mobile application (see Subsection 3.4) and conduct a large-scale user study to get real feedback.

Inferred Seasonal Popularity.

The inferred seasonal popularity performed best among all our inferred parameters. An excerpt of the POI data with seasonal popularity is shown in Table 2. While cities like Bad Tölz or Nürnberg are equally popular throughout the year, other sights peak in certain seasons. Kehlsteinhaus for instance - a tea house built for the Third Reich government that has been turned into an exhibition - is closed during spring and winter (November - April), as the road can not be maintained as soon as there is snowfall. It can be easily seen that this shutdown persists in the data making the sight less popular during winter and spring. Other places for outdoor activities show a clear tendency towards the warmer periods of the year, including Walchensee (being popular during Summer and Fall), Starnberger See (Summer) and Königssee (Summer and Fall).

POI Name	Spring	Summer	Fall	Winter
Bad Tölz	0.215	0.202	0.215	0.368
Nürnberg	0.238	0.242	0.295	0.226
Walchensee	0.076	0.456	0.291	0.177
Kehlsteinhaus	0.044	0.572	0.35	0.034
Starnberger See	0.003	0.932	0.045	0.019
Kloster Andechs	0.314	0.253	0.365	0.067
Königssee	0.193	0.367	0.305	0.135

Table 2: Excerpt of Inferred Seasonal Popularity: $p(V|s,P)$**.** $\mu = 1,660$ **Check-ins**

Inferred Weather Condition Popularity.

The weather condition overall has a clear tendency towards sunny weather, with more then 90% of the dataset having a sunny popularity of 0.5 or higher. Nonetheless, there are still some patterns to get indications if the inference technique is correct. While cities like Munich are within our average (mostly sunny), places for outdoor activities like Starnberger See, Andechs or Almbachklamm have a notably higher sunny popularity. Table 3 shows an excerpt from inferred weather condition popularities.

POI Name	Sunny	Cloudy	Rainy	Snowy	Unkn.
Munich	0.532	0.138	0.324	0.005	0.001
Andechs	0.711	0.072	0.209	0.006	0.002
Roseninsel	0.796	0.127	0.072	0.005	0
Almbachkl.	0.907	0	0.093	0	0

Table 3: Excerpt of Inferred Weather Condition Popularity: $p(V|wc, P)$. $\mu = 1,750$ **Check-ins**

5. CONCLUSIONS AND FUTURE WORK

In this work, we have designed and implemented a data analytics process to infer the popularity of POIs for a context-aware recommender system using publicly accessible data from various data sources. We have elaborated different approaches on individual subtasks, e.g., applying probabilistic modeling to learn the popularity for POIs in different contextual situations such as *weather condition, day of week* and *time of day*. We qualitatively evaluated our approach in a case study and also implemented a corresponding mobile application with a brief preliminary user study [7].

In addition to the explained results, we discovered other useful purposes for the dataset and the inferred information. For example, to determine and visualize popular areas on a map. This could be useful for visitors to find spots for taking iconic pictures or maybe find a less-crowded spot and avoid the most popular areas. To do so, we filtered our data by first selecting the check-ins associated with a POI and then applying another filter to eliminate check-ins with an accuracy worse then 10 meters. Then this filtered dataset is fed into a heatmap algorithm for visualization using Google Maps' heatmaps extension. Figure 5 shows popular (photo) spots around Neuschwanstein Castle as an example.

Future work includes integrating the item popularities into the mobile tourist guide application and proactively recommending POIs based on the current user context. We then plan to conduct a larger user study to investigate whether this approach leads to useful and timely recommendations from a user's perspective in a real setting. Another area for improvement is not to solely rely on explicit user check-ins but also utilize tracking data from smartphones.

Figure 5: Heatmap for Neuschwanstein Castle Based on Check-in Data

6. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 191–226. Springer, 2015.

[2] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, 16(5):507–526, 2012.

[3] J. Borrís, A. Moreno, and A. Valls. Review: Intelligent tourism recommender systems: A survey. *Expert Syst. Appl.*, 41(16):7370–7389, Nov. 2014.

[4] M. Braunhofer, M. Elahi, and F. Ricci. Usability assessment of a context-aware and personality-based mobile recommender system. In *E-commerce and web technologies*, pages 77–88. Springer, 2014.

[5] N. Dali Betzalel, B. Shapira, and L. Rokach. Please, not now!: A model for timing recommendations. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 297–300. ACM, 2015.

[6] M. Daszykowski and B. Walczak. Density-Based Clustering Methods. *Comprehensive Chemometrics*, 2:635–654, 2010.

[7] P. Hiesel, W. Wörndl, M. Braunhofer, and D. Herzog. A User Interface Concept for Context-Aware Recommender Systems. In *Mensch und Computer 2016 Tagungsband*. De Gruyter, 2016.

[8] J. Melià-Seguí, R. Zhang, and E. Bart. Activity duration analysis for context-aware services using foursquare check-ins. In *Proceedings of the 2012 international workshop on Self-aware internet of things*, pages 13–18, 2012.

[9] M. Pielot, L. Baltrunas, and N. Oliver. Boredom-triggered proactive recommendations. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pages 1106–1110. ACM, 2015.

[10] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer Verlag, New York, NY, USA, 2010.

[11] T. A. Runkler. *Data Analytics: Models and Algorithms for Intelligent Data Analysis*. Springer, 2012.

[12] G. D. Tre and A. Bronselaer. Consistently handling geographical user data: Merging of coreferent POIs. In *Proceedings of the 2010 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, 2010.

[13] D. G. Vico, W. Wörndl, and R. Bader. A study on proactive delivery of restaurant recommendations for android smartphones. In *ACM RecSys workshop on personalization in mobile applications, Chicago, USA*, 2011.

[14] W. Wörndl, J. Hübner, R. Bader, and D. Gallego-Vico. A model for proactivity in mobile, context-aware recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 273–276. ACM, 2011.

Understanding the Impact of Weather for POI Recommendations

Christoph Trattner*
Know-Center, Austria

Alexander Oberegger
TUG, Austria

Lukas Eberhard
TUG, Austria

Denis Parra
PUC, Chile

Leandro Marinho
UFCG, Brasil

ABSTRACT

POI recommender systems for location-based social network services, such as Foursquare or Yelp, have gained tremendous popularity in the past few years. Much work has been dedicated into improving recommendation services in such systems by integrating different features that are assumed to have an impact on people's preferences for POIs, such as time and geolocation. Yet, little attention has been paid to the impact of weather on the users' final decision to visit a recommended POI. In this paper we contribute to this area of research by presenting the first results of a study that aims to recommend POIs based on weather data. To this end, we extend the state-of-the-art Rank-GeoFM POI recommender algorithm with additional weather-related features, such as temperature, cloud cover, humidity and precipitation intensity. We show that using weather data not only significantly increases the recommendation accuracy in comparison to the original algorithm, but also outperforms its time-based variant. Furthermore, we present the magnitude of impact of each feature on the recommendation quality, showing the need to study the weather context in more detail in the light of POI recommendation systems.

Keywords

POI Recommender Systems; Location-based services; Weather-Context

1. INTRODUCTION

Location-based social networks (LBSN) enable users to check-in and share places and relevant content, such as photos, tips and comments that help other users in exploring novel and interesting places in which they might not have been before. Foursquare[1], for example, is a popular LBSN with millions of subscribers doing millions of check-ins everyday all over the world[2]. This vast amount of check-in data, publicly available through Foursquare's data access APIs, has inspired many researchers to investigate human mobility patterns and behaviors with the aim of assisting users by means of personalized POI (point of interest) recommendation services [15, 16].

Problem Statement. The problem we address in this paper is the POI recommendation problem. Hence, given a user u and their check-in history L^u, i.e., the POIs that they have visited in the past, and current weather conditions $C = \{c_1, \ldots, c_{|C|}\}$, where c_i are weather features such as temperature, wind speed, pressure, etc., we want to recommend the POIs $\hat{L}^u = \{l_1, \ldots, l_{|L|}\}$ that they will likely visit in the future that are not in L^u.

Objective. Most of the existing approaches on POI recommendation exploit three main factors (aka contexts) of the data, namely, social, time and geolocation [5, 10, 15]. While these approaches work reasonably well, little attention has been paid to weather, a factor that may potentially have a major impact on users' decisions about visiting a POI or not. For example, if it is raining in a certain place in a certain period of time, the user may prefer to check-in indoor POIs.

In this paper we contribute to this area of research by presenting the first results of a recently started project that exploits weather data to recommend, for a given user within a given city, the POIs that they will likely visit in the future. To this end, we extract several weather features based on data collected from forecast.io such as temperature, cloud cover, humidity or precipitation intensity, and feed it into a state-of-the-art POI recommender algorithm called Rank-GeoFM [10]. The reason why we decided to build our approach on top of this algorithm is twofold: (i) Rank-GeoFM has shown to outperform other strong baselines from the literature and (ii) it is very easy to extend it with additional contextual data.

Research Questions. To drive our research the following three research questions were defined:

- **RQ1.** Do weather conditions have a relation with the check-in behavior of Foursquare users?
- **RQ2.** Is it possible to improve current POI recommendation quality using these weather features?
- **RQ3.** Which weather features provide the highest impact on the recommendations?

*Corresponding author: ctrattner@know-center.at
[1]https://foursquare.com/

[2]https://foursquare.com/infographics/10million

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

City	#Check-Ins	#Venues	#Users	Sparsity
Minneapolis	37,737	797	436	89.1%
Boston	42,956	1141	637	94.3%
Miami	29,222	796	410	91.0%
Honolulu	16,042	410	173	77.4%

Table 1: Basic statistics of the dataset.

Contributions. To the best of our knowledge, this is the first paper that investigates in detail the extent to which weather features such as temperature, cloud cover, humidity or precipitation intensity impact on users' check-in behaviors and how these features perform in the context of POI recommender systems. Although there is literature showing that POI recommender systems can be improved by using some kind of weather context such as e.g. temperature, it is not clear yet, how much they add or what type of weather feature is the most/least useful one. Another contribution of this paper is the introduction of a weather-aware recommender method that builds upon a very strong state-of-the-art POI recommender system called Rank-GeoFM. The method is implemented and embedded into the very popular recommender framework MyMediaLite [7] and can be downloaded for free from our GitHub repository (details in Section 8).

Outline. The structure of this paper is as follows: In Section 2 we highlight relevant related work in this field. Section 3 describes how we enriched Rank-GeoFM with weather data. Section 4 describes the experimental setup and presents results from our empirical analysis. Section 5 presents insights on the results obtained with our weather-aware recommender approach. Finally, Sections 6 and 7 conclude the paper with a summary of our main findings and future directions of the work.

2. RELATED WORK

With the advent of LBSNs, POI recommendation rapidly became an active area of research within the recommender systems, machine learning and Geographic Information Systems research communities [2]. Most of the existing research works in this area exploit some sort of combination between some (or all) of the following data sources: check-in history, social relations (e.g. friendship relations), time and geolocations [1, 5, 6, 8, 10, 13, 15]. While these different sources of data (aka contexts) affect the user's decision on visiting a POI in different ways, weather data, which according to common sense may have a great influence on this decision, are still rarely used.

Martin et al. [11] proposed a mobile application which architecture considered the use of weather data to personalize a geocoding mobile service, but no implementation or evaluation was presented. A similar contribution was done by Meehan et al. [12], who proposed a hybrid recommender system based on time, weather and media sentiment when introducing the VISIT mobile tourism recommender, but they neither implemented nor evaluated it.

Among the few works that have actually used weather in the recommendation pipeline, Braunhofer et al. [3] introduced a recommender system designed to run in mobile applications for recommending touristic POIs in Italy. The authors conducted an online study with 54 users and found out that recommendations that take into consideration weather information were indeed able to increase the user satisfaction. Compared to this work, our implementation is based in a more recent and state-of-the-art algorithm, and we also provide details of which weather features contribute the most to the recommender performance. In an extension of their initial work, Braunhofer et al. [4] implemented and evaluated a context-aware recommender system which uses weather data. They find that the model which leverages the weather context outperformed the version without it. Although more similar to our current work, they did not provide a detailed feature analysis as the present article.

In summary, compared to previous works which have used weather as a contextual factor for recommendation systems, we provide detailed information about the recommendation algorithm and we contribute an implementation extending a state-of-the-art matrix factorization model exploiting rich weather data. Moreover, we also provide details on how the weather features were exploited by it, as well as a detailed analysis about the impact of the features on the recommendation performance.

Sym.	Description
$\mathcal{U}$	set of users $u_1, u_2, ..., u_{\|U\|}$
$\mathcal{L}$	set of POIs $l_1, l_2, ..., l_{\|L\|}$
FC_f	set of classes for feature f
F	set of weather feature classes $f_1, f_2, ..., f_{\|FC_f\|}$
Θ	latent model parameters containing the learned weights $\{L^{(1)}, L^{(2)}, L^{(3)}, U^{(1)}, U^{(2)}, F^{(1)}\}$ for locations, users and weather features.
X_{ul}	$\|U\| \times \|L\|$ matrix containing the check-ins of users at POIs.
X_{ulc}	$\|U\| \times \|L\| \times \|FC_f\|$ matrix containing the check-ins of users at POIs at a specific feature class c.
D_1	user-POI pairs: $(u,l)\|x_{ul} > 0$.
D_2	user-POI-feature class triples: $(u,l,c)\|x_{ulc} > 0$.
W	geographical probability matrix of size $\|L\|x\|L\|$ where $w_{ll'}$ contains the probability of l' being visited after l has been visited according to their geographical distance. $w_{ll'} = (0.5 + d(l,l'))^-1)$ where $d(l,l')$ is the geographical distance between the latitude and longitude of l and l'.
WI	probability that a weather feature class c is influenced by feature class c'. $wi_{cc'} = cos\_sim(c,c')$.
$N_k(l)$	set of k nearest neighbors of POI l.
y_{ul}	the recommendation score of user u and POI l.
y_{ulc}	the recommendation score of user u, POI l and weather feature class c.
$I(\cdot)$	indicator function returning $I(a) = 1$ when a is true and 0 otherwise.
ϵ	margin to soften ranking incompatibility.
γ_w	learning rate for updates on weather latent parameters.
γ_g	learning rate for updates on latent parameters from base approach.
$E(\cdot)$	a function that turns the rating incompatibility $Incomp(y_{ulc}, \epsilon)$, that counts the number of locations $l' \in \mathcal{L}$ that should be ranked lower than l at the current weather context c and user u but are ranked higher by the model, into a loss $E(r) = \sum_{i=1}^{r} \frac{1}{i}$.
$\delta_{ucll'}$	function to approximate the indicator function with a continuous sigmoid function $s(a) = \frac{1}{1+exp(-a)}$. $\delta_{ucll'} = s(y_{ul'c} + \epsilon - y_{ulc})(1 - s(y_{ul'c} + \epsilon - y_{ulc}))$
$\lfloor \frac{\|L\|}{n} \rfloor$	if the n^{th} location l' was ranked incorrect by the model the expactation is that overall $\lfloor \frac{\|L\|}{n} \rfloor$ locations are ranked incorrect.
g, μ	auxiliary variable that save partial results of the calculation of the stochastic gradient.

Table 2: The notations used to describe Rank-GeoFM and the incorporation of the weather context.

3. RECOMMENDATION APPROACH

Our recommendation approach is built upon a state-of-the-art POI recommender algorithm named Rank-GeoFM [10], a personalized ranking based matrix factorization method.

Algorithm 1: Rank-GeoFM with weather context

Input: check-in data D_1, D_2, geographical influence matrix W, weather influence matrix WI, hyperparameters ϵ, C, α, β and learning rate γ_g and γ_w
Output: parameters of the model $\Theta = \{L^{(1)}, L^{(2)}, L^{(3)}, U^{(1)}, U^{(2)}, F\}$

1 **init:** Initialize Θ with $\mathcal{N}(0, 0.01)$; Shuffle D_1 and D_2 randomly
2 **repeat**
3   **for** $(u, l) \in D_1$ **do**
4   approach from Li et al. [10]
5   **end**
6   **for** $(u, l, c) \in D_2$ **do**
7   Compute y_{ulc} as Equation 3 and set n = 0
8   **repeat**
9   Sample l' and c', Compute $y_{ul'c'}$ as Equation 3
10   n++
11   **until** $I(x_{ulc} > x_{ul'c'})I(y_{ulc} < y_{ul'c'} + \epsilon) = 1$ *or* $n > |L|$
12   **if** $I(x_{ulc} > x_{ul'c'})I(y_{ulc} < y_{ul'c'} + \epsilon) = 1$ **then**
13             $\eta = E\left(\left\lfloor \frac{|L|}{n} \right\rfloor\right) \delta_{ucll'}$
14             $g = \left(\sum_{c^* \in FC_f} wi_{c'c^*} f^{(1)}_{c*} - \sum_{c^+ \in FC_f} wi_{cc^+} f^{(1)}_{c+}\right)$
15             $f^{(1)}_c \leftarrow f^{(1)}_c - \gamma_w \eta (l^{(2)}_{l'} - l^{(2)}_l)$
16             $l^{(3)}_l \leftarrow l^{(3)}_l - \gamma_w \eta g$
17             $l^{(2)}_{l'} \leftarrow l^{(2)}_{l'} - \gamma_w \eta f_c$
18             $l^{(2)}_l \leftarrow l^{(2)}_l + \gamma_w \eta f_c$
19   **end**
20   Project updated factors to accomplish constraints
21   **end**
22 **until** *convergence*
23 **return** $\Theta = \{L^{(1)}, L^{(2)}, L^{(3)}, U^{(1)}, U^{(2)}, F^{(1)}\}$

We have selected Rank-GeoFM over other alternatives, because it has been shown to be a very strong POI recommender method compared to other approaches often cited in the literature. In Li et al. [10] the authors compared Rank-GeoFM against twelve other recommender methods, showing that Rank-GeoFM significantly outperforms strong generic baselines, such as user-KNN, item-KNN CF, WRMF, BPR-MF [7] as well as specialized POI recommender methods, such as BPP [17]. Another reason for choosing Rank-GeoFM is related to its ability to easily accommodate additional features, such as the ones that we plan to use in this work. The aim of Rank-GeoFM is to learn latent parameters that model the relationship between the context of interest (in our case weather conditions) and the user/POI.

Table 2 describes the symbols used in the recommender algorithm. For each type of contextual data considered, latent model parameters are introduced. The prediction score of a <*user, POI, context*> triple is then made based on this learned latent parameters. The parameters are trained using a fast learning scheme introduced by the authors that is based on Stochastic Gradient Descent (SGD).

To add the weather context into Rank-GeoFM, the weather features' values needed to be discretized. This was done to reduce data sparsity. For example, if we considered temperature as a real number, most of the check-ins concerning specific temperature values would probably be zero. Thus, transforming continuous values of weather features (e.g., temperature) into intervals might alleviate this problem. Hence, a mapping function is introduced (see Equation 1) that converts the weather features into interval bins. $|FC_f|$ defines the number of bins for the current weather feature. We will refer to these bins as feature classes. The best results were obtained with $|FC_f| = 20$ (validated on hold-out data).

$$c_f(value) = \left\lfloor \frac{(value - min(f)) \cdot (|FC_f| - 1)}{(max(f) - min(f))} \right\rfloor \qquad (1)$$

To extend the original Rank-GeoFM approach with weather context, three additional latent factors are introduced that are represented by matrices in a K-dimensional space. The first one is for incorporating the *weather-popularity-score* that models whether or not a location is popular with respect to a specific weather feature class and is named $L^{(2)} \in \mathbb{R}^{|L| \times K}$, where K denotes the size of the latent parameter space. Furthermore, a matrix $L^{(3)} \in \mathbb{R}^{|L| \times K}$ is introduced to model the *influence between two feature classes*. In other words, $L^{(3)}$ softens the borders between the particular feature classes. The third latent parameter $F^{(1)} \in \mathbb{R}^{|FC_f| \times K}$ is then used to parametrize the feature classes of the specific weather feature. In addition to the latent parameters, a Matrix $WI \in \mathbb{R}^{|FC_f| \times |FC_f|}$ is introduced for storing the probability that a weather feature class c is influenced by feature class c'. Denoting x_{ulc} as the frequency that a user u checked-in POI l with the current weather context c, this probability is calculated as follows:

$$wi_{cc'} = \frac{\sum_{u \in U} \sum_{l \in L} x_{ulc} x_{ulc'}}{\sqrt{\sum_{u \in U} \sum_{l \in L} x^2_{ulc}} \sqrt{\sum_{u \in U} \sum_{l \in L} x^2_{ulc'}}} \qquad (2)$$

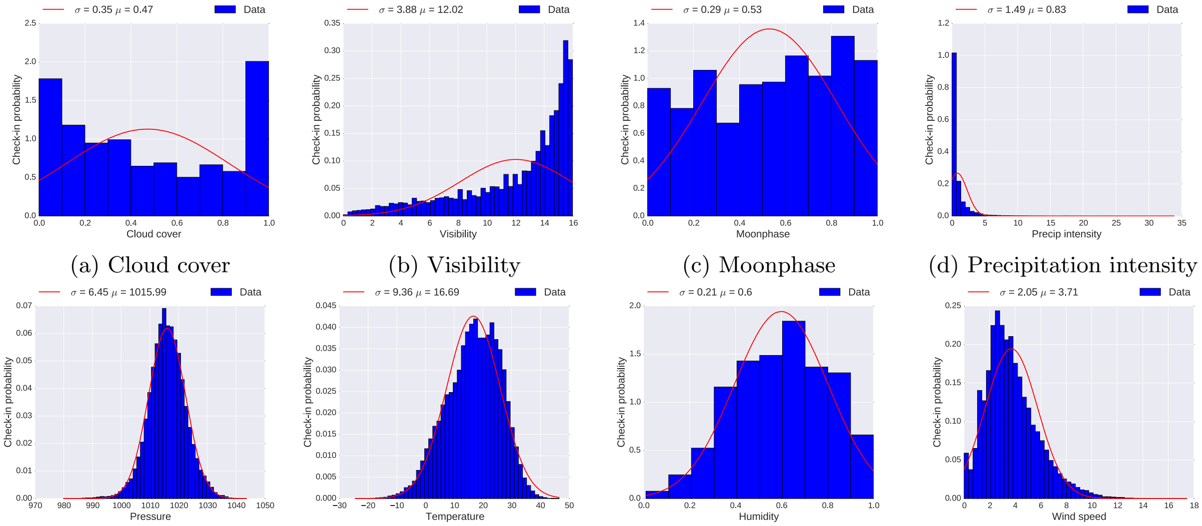
To calculate the recommendation score for a given user u, POI l and weather feature class c, Equation 3 is introduced, where y_{ul} denotes the recommendation score as computed in Li et al. [10].

$$\begin{aligned} y_{ul} &= u^{(1)}_u \cdot l^{(1)}_l + u^{(2)}_u \cdot \sum_{l^* \in N_k(l)} w_{ll^*} l^{(1)}_{l^*} \\ y_{ulc} &= y_{ul} + f^{(1)}_c \cdot l^{(2)}_l + l^{(3)}_l \cdot \sum_{c^* \in FC} wi_{cc^*} f^{(1)}_{c^*} \end{aligned} \qquad (3)$$

Algorithm 1 describes how we incorporated the weather context features into the base Rank-GeoFM approach. Taking the initialization and the hyperparameters from the original approach, we first iterate over all pairs of users and POIs $(u, l) \in D_1$, where D_1 is the set of all check-ins and do the adjustments of the latent parameters as described in Li et al. [10].

We then introduce an iteration over all <*user, venue, feature-class*> triples $(u, l, c) \in D_2$ in order to adjust the latent parameters on the incorrect ranked venues according to the specific weather context. This adjustment is necessary because the algorithm might rank a triple (u, l, c) correctly where on the other hand (u, l, c') might be ranked incorrectly. The adjustments are then done accordingly to the base algorithm in lines 6-20.

During our studies we found that with a learning rate of $\gamma_g = .0001$, as used in Li et al. [10], the algorithm did not converge. The reason for that is that the adjustments are done on a higher granularity for each (u, l, c) triple and not just on the (u, l) level. Henceforth, we introduce a new learning rate parameter $\gamma_w = .00001$ for the weather con-



(a) Cloud cover (b) Visibility (c) Moonphase (d) Precipitation intensity

(e) Pressure (f) Temperature (g) Humidity (h) Windspeed

Figure 1: Check-in distributions over the eight weather features.

(a) "Austrian Restaurant" (b) "Farm"

(c) "Ski Area" (d) "Ice Cream Shop"

Figure 2: Examples of check-in distributions over different types of places in Foursquare. On the left hand side, places where people check-in at lower temperatures are shown and on the right higher temperature places are featured.

text, for which stable results could be observed (validation on hold-out data). Similarly to Li et al. [10], we found in our experiments that the best values of the hyperparameters are as follows (validated on hold-out data): $\epsilon = .3$, $C = 1.0$, $\alpha = \beta = .2$, and $K = 100$ as used for the dimensions of the matrices $L^{(1)}$, $L^{(2)}$ and $L^{(3)}$.

4. EXPERIMENTAL SETUP

In this section we describe in detail our experimental setup, i.e., the datasets we used, a brief characterization of these datasets concerning the weather features used, and the evaluation protocol we have chosen to conduct our study.

4.1 Datasets

The dataset we used in this study was obtained from the work of Yang et al. [14]. It is a Foursquare crawl comprising user check-in data from April 2012 to September 2013. The original dataset contains more than 33 million check-ins from 415 cities in 77 countries. However, before dealing with our problem on such a large scale, we decided to first concentrate our investigation on a small set of US cities. We selected four cities that could represent some weather variety in order to investigate whether our model is robust to such variety of weather conditions (see Figure 3). Table 1 provides an overview of the check-in statistics of the four target cities chosen for our experiments: Minneapolis, Boston, Miami and Honolulu.

Concerning the weather information, we have used the API of forecast.io[3] to collect, for each $<$*time, place*$>$ tuple present in our dataset, their corresponding weather information. For that, we need to pass the following request to the API:

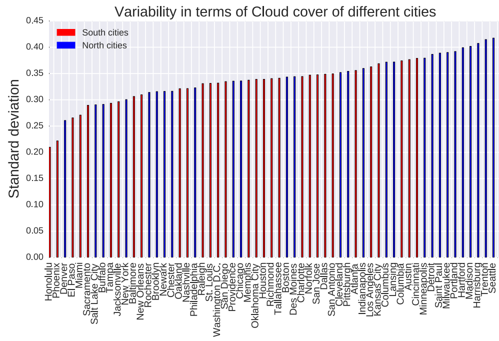
https://api.forecast.io/forecast/APIKEY/LAT,LON,TIME

For the purposes of our analysis, we obtained eight weather features, namely, cloud cover, visibility, moon phase, precipitation intensity, pressure, temperature, humidity and wind speed, for all places and time-stamps in our dataset that are provided by forecast.io.

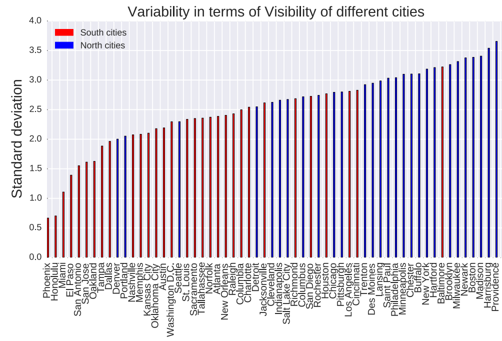
4.2 Data Analysis

Figure 1 shows the probability distributions of check-ins for each of the eight weather features used. Notice that the distributions of pressure, temperature, humidity and wind speed resemble a normal distribution (see the colored approximation curve). Moreover, while moon phase seems to follow a uniform distribution, which indicates that it will likely not help the recommendation model, the distribution of precipitation is very skewed, showing that users have a strong preference to check-in places when there is low precipitation intensity (i.e., not raining), indicating that this feature might have a good discriminative power.

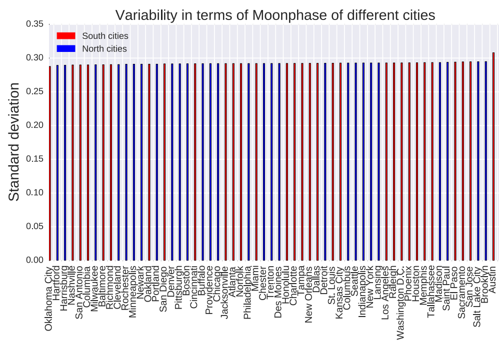
[3]https://developer.forecast.io/docs/v2



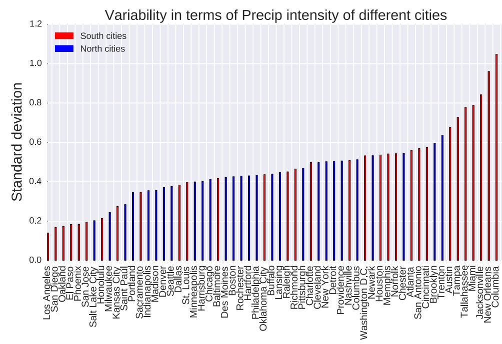
(a) Cloud cover



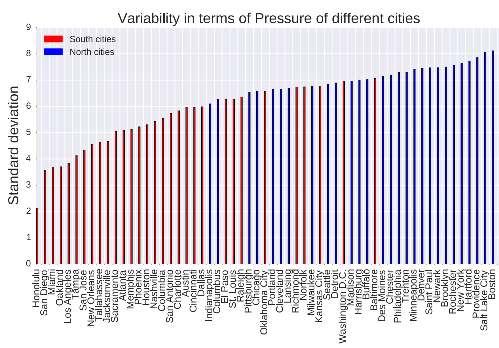
(b) Visibility



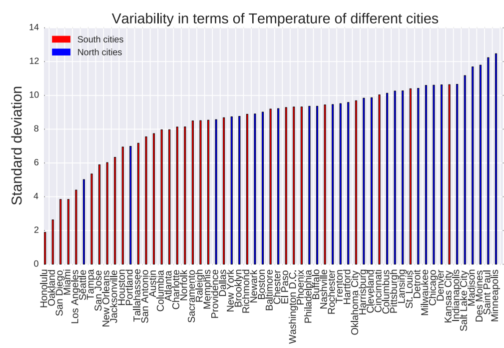
(c) Moonphase



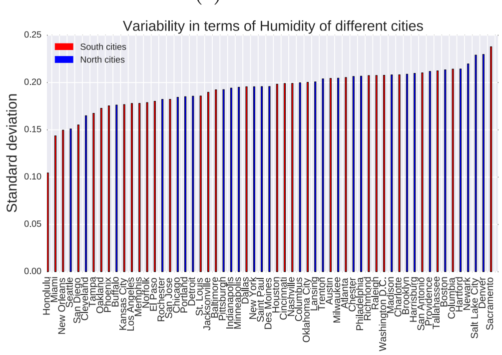
(d) Precipitation intensity



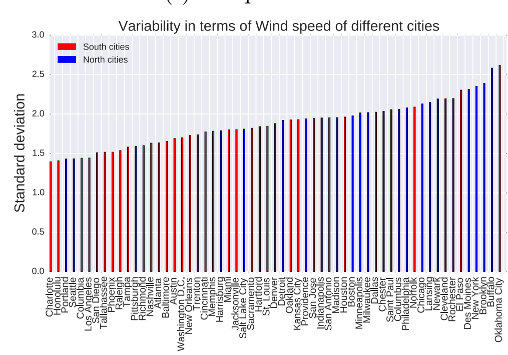
(e) Pressure



(f) Temperature

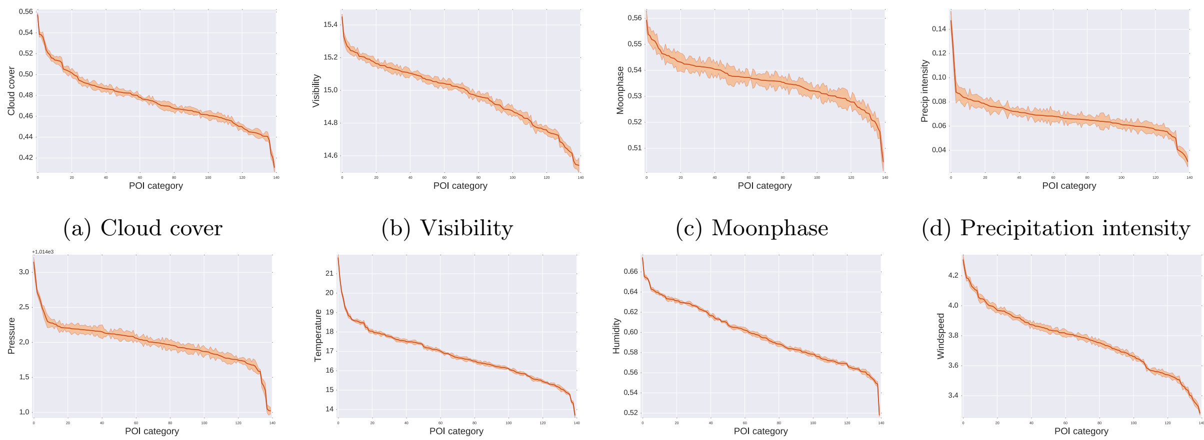


(g) Humidity



(h) Windspeed

Figure 3: Weather feature variability (sorted) measured via standard deviation over cities. Left: cities with lowest variability. Right: cities with highest variability.



(a) Cloud cover (b) Visibility (c) Moonphase (d) Precipitation intensity

(e) Pressure (f) Temperature (g) Humidity (h) Windspeed

Figure 4: Mean weather feature values (sorted) for POI categories with standard errors.

In addition to this, Figure 2 illustrates the check-in distribution as a function of temperature in four different POI categories. As highlighted in this figure, different patterns occur depending on the category chosen. While people prefer to check-in in e.g., “Austrian Restaurants” or “Ski Areas” when the temperature is low, “Ice Cream Shops” or “Farms” are preferred when temperatures are higher.

Figure 3 shows how the weather features vary in each city of the original Foursquare dataset. Notice that with the exception of moon phase, all the features present a dependency regarding the city where they are measured, indicating that a different recommendation model should probably be trained for each different city. Moreover, in general, weather shows a higher variability in the north of the US and a very low variability in the south that peaks in the island Honolulu which shows almost no variability in terms of weather. Figure 4 shows the different mean values of the eight weather features over the POI categories. With the small overlapping of the standard error of the means it’s revealed that indeed categories have a distinct popularity across various weather feature values. Even moon phase shows a divergent category popularity at its tails.

After this analysis we can confidently state that there is indeed a relation between the weather conditions and the check-in behavior of Foursquare users, which answers our first research question (RQ1) stated at Section 1.

4.3 Evaluation

Protocol. To evaluate the performance of our algorithm, we have chosen the same evaluation protocol as described in the original Rank-GeoFM paper [10]. Hence, we split the dataset (according to the time line) into training, validation and test sets for each city by adding the first 70% of the check-ins of each user to the training set, the following 20% to the test set and the rest to the validation set (=10%). The training set was then used to learn the latent model parameters. During the training phase of the algorithm, the validation set was used to tune the algorithm convergence. When convergence was observed (typically around 3,000 – 5,000 iterations with fast learning scheme enabled), the training was stopped and the learned parameters were used to evaluate the model on the test set.

Baselines. As baselines for our experiments, we used the original Rank-GeoFM approach, that takes into account both the check-in history of users and geographical influence. We also compare to the time-based method of Rank-GeoFM, that was also introduced in Li et al. [10].

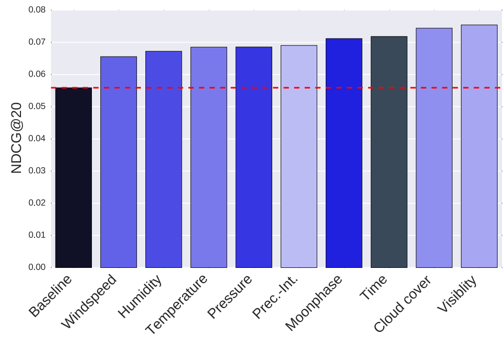
Metric. As evaluation metric NDCG@k (Normalized Discounted Cumulative Gain) with $k = 20$[4] was chosen, as we want to predict the top-k POIs for a user.

5. RESULTS

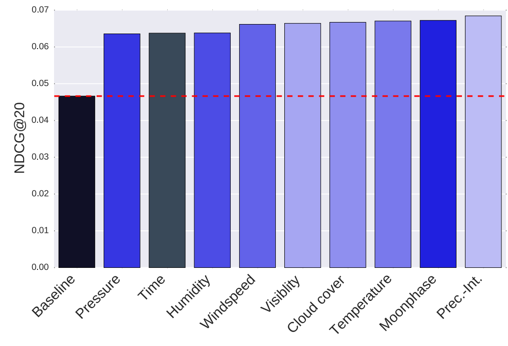
Figure 5 shows the results of our offline experiment. As shown, in all cases Rank-GeoFM enriched with our proposed weather features significantly outperforms the original Rank-GeoFM algorithm, which answers RQ2. For all pairwise-comparisons (recommenders with weather context vs. without) a standard t-test showed that the p-values were always smaller than $p < .001$. What is even more interesting to note is the performance of Rank-GeoFM that utilizes the time feature as contextual factor. As highlighted, in all cases, Rank-GeoFM with weather features, such as visibility and precipitation intensity outperforms the time-based variant, showing that indeed weather conditions may help to improve the recommendation quality.

We also highlight the fact that certain weather features perform better than others and this ranking seems to be city dependent. This can be clearly observed in Figure 5, where the results of Rank-GeoFM with each weather feature is shown. This answers RQ3, showing which features provide the highest gain in recommendation quality. For example, in Honolulu the best performing feature is precipitation intensity, while in Minneapolis visibility seems to work best among all investigated weather features. Similar patterns can be observed for other features, such as temperature or cloud cover, changing their relative importance across the four cities. These observations are in line with the results in Figure 1, showing a strong tendency of check-ins into POIs under certain weather conditions. However, what is also interesting to note is the good performance of the moon phase feature, which appeared to be uniformly distributed in gen-

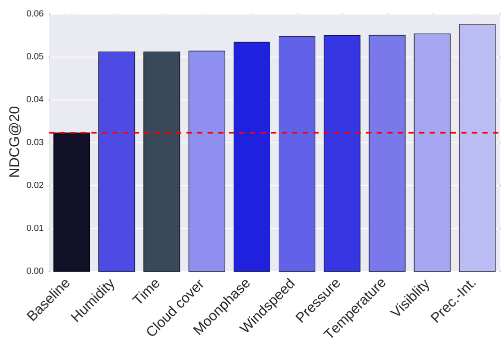
[4]Please note, that we have also run simulations with $k = 5$ or 10, with similar trends in the results as obtained with $k = 20$. However, due to limited space, they were not included into this paper.



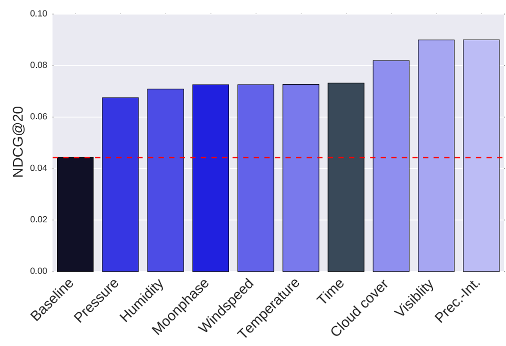
(a) Minneapolis



(b) Boston



(c) Miami



(d) Honolulu

Figure 5: Recommender accuracy for the eight different weather context features (sorted by importance) compared to Rank-GeoFM without weather context (denoted as “Baseline”). For further comparison the time-aware version of Rank-GeoFM is included, denoted as “Time”. The red dotted line denotes the baseline.

eral (cf. Figure 1). Hence, it appears, that at the level of locations there is indeed a strong preference for check-ins in different phases of the moon. In a recent research, Kohyama et a. [9] found a relation between moon phase, tidal variation, humidity and rainfall. Notably, we found a positive relation by analyzing these data based on check-ins, finding a small but positive correlation between moon phase and precipitation intensity, humidity, cloud cover and pressure, as seen in the last row of the correlation matrix shown in Figure 6. Although further analysis should be performed to establish a link between our study and theirs, this might be a possible explanation regarding the effect of moon phase in our POI recommendation model.

Finally, the relative performance improvement over the original Rank-GeoFM also seems to be location dependent. Hence, while our approach work to a great extent better compared to the baseline for Miami and Honolulu, the differences are less pronounced for Minneapolis. One reason for this observation could be that there are more POIs available showing similar weather profiles. However, to further confirm these hypotheses, additional analyses are needed.

6. CONCLUSIONS

In this paper we presented our preliminary findings on how weather data may affect users’ check-in behavior and how this information can be used in the context of a POI recommender system. As our preliminary analyses on the Foursquare check-in data showed, the weather factors have indeed a significant impact on the people’s check-in behavior, showing different check-in profiles for different kinds of places (which answers RQ1). Furthermore, we use the proposed weather features within a state-of-the-art POI recommender and we were able to increase the recommender accuracy in comparison to the original method that does not use weather data (thus answering RQ2). Furthermore, our experiments revealed that the weather context is more useful than the context of time and, that the weather features used in this work are city-dependent. Finally, our study showed (see RQ3) that among the considered weather features, precipitation intensity and visibility are the most significant ones to improve the ranking in a weather-aware POI recommender system.

7. FUTURE WORK

Currently, our work only investigates one weather feature at a time. Investigating different hybridization or context-aware recommender system (CARS) methods and other context variables will be therefore a task to be conducted in our future work. Furthermore, it will help to investigate in more detail, how the algorithm performs on the whole Foursquare dataset, as more interesting patterns across cities may occur. Finally, we would like to extend our investigations also at user levels, since the current ones concentrate only on the weather profiles of the POIs.

8. OPEN SCIENCE

In order to make the results obtained in this work reproducible, we share code and data of this study. The proposed method Rank-GeoFM with weather context is implemented

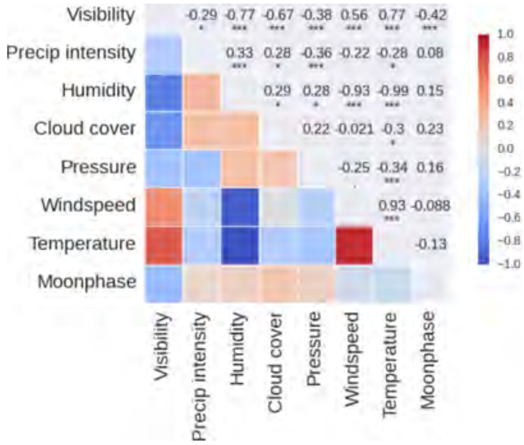


Figure 6: Correlation matrix for the eight weather features investigated (*$p < 0.5$, **$p < 0.01$, ***$p < 0.001$).

with the help of the MyMediaLite framework [7] and can be downloaded for free from our GitHub repository[5]. Furthermore, the data samples used in the experiments can be requested for free via email to the corresponding author.

Acknowledgements

This work is supported by the Know-Center. The Know-Center is funded within the Austrian COMET Program - managed by the Austrian Research Promotion Agency (FFG). The authors Denis Parra and Leandro Marinho were supported by CONICYT, project FONDECYT 11150783 and EU-BR BigSea project (MCTI/RNP 3rd Coordinated Call) respectively.

9. REFERENCES

[1] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 199–208, New York, NY, USA, 2012. ACM.

[2] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel. Recommendations in location-based social networks: A survey. *Geoinformatica*, 19(3):525–565, July 2015.

[3] M. Braunhofer, M. Elahi, M. Ge, F. Ricci, and T. Schievenin. STS: design of weather-aware mobile recommender systems in tourism. In *Proceedings of the First International Workshop on Intelligent User Interfaces: Artificial Intelligence meets Human Computer Interaction (AI*HCI 2013) A workshop of the XIII International Conference of the Italian Association for Artificial Intelligence (AI*IA 2013), Turin, Italy, December 4, 2013.*, 2013.

[4] M. Braunhofer, M. Elahi, F. Ricci, and T. Schievenin. Context-aware points of interest suggestion with dynamic weather data management. In *Information and communication technologies in tourism 2014*, pages 87–100. Springer, 2014.

[5] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proc. of AAAI*, pages 17–23, 2012.

[6] G. Ference, M. Ye, and W.-C. Lee. Location recommendation for out-of-town users in location-based social networks. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 721–726, New York, NY, USA, 2013. ACM.

[7] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A free recommender system library. In *In Proc. of RecSys'11*, 2011.

[8] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 93–100, New York, NY, USA, 2013. ACM.

[9] T. Kohyama and J. M. Wallace. Rainfall variations induced by the lunar gravitational atmospheric tide and their implications for the relationship between tropical rainfall and humidity. *Geophysical Research Letters*, 43(2):918–923, 2016. 2015GL067342.

[10] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *Proc. of SIGIR'15*, pages 433–442, New York, NY, USA, 2015. ACM.

[11] D. Martin, A. Alzua, and C. Lamsfus. *A Contextual Geofencing Mobile Tourism Service*, pages 191–202. Springer Vienna, Vienna, 2011.

[12] K. Meehan, T. Lunney, K. Curran, and A. McCaughey. Context-aware intelligent recommendation system for tourism. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 328–331. IEEE, 2013.

[13] I. Nunes and L. Marinho. A personalized geographic-based diffusion model for location recommendations in lbsn. In *Proceedings of the 2014 9th Latin American Web Congress*, LA-WEB '14, pages 59–67, Washington, DC, USA, 2014. IEEE Computer Society.

[14] D. Yang, D. Zhang, and B. Qu. Participatory cultural mapping based on collective behavior in location based social networks. *ACM Transactions on Intelligent Systems and Technology*, 2015. in press.

[15] M. Ye, P. Yin, W.-C. Lee, and D. L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proc. of SIGIR'11*, pages 325–334. ACM, 2011.

[16] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: a location-content-aware recommender system. In *Proc. of KDD'13*, pages 221–229. ACM, 2013.

[17] Q. Yuan, G. Cong, and A. Sun. Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proc. of CIKM'14*, pages 659–668. ACM, 2014.

[5]https://github.com/aoberegg/WPOI

A Motivation-Aware Approach for Point of Interest Recommendations

Khadija Ali Vakeel
Indian Institute of Management Indore
Prabandh Shikhar, Indore
India
091-0731-2439-666
f13khadijav@iimidr.ac.in

Sanjog Ray
Indian Institute of Management Indore
Prabandh Shikhar, Indore
India
091-0731-2439-524
sanjogr@iimidr.ac.in

ABSTRACT
Most existing context aware recommender systems primarily use a combination of ratings data, content data like features or attributes of the product or service, context data like location or time and social network data. In this paper, we propose a novel approach for refining the recommendations made by location-aware recommender systems based on user motivations for checking in at locations in location based social networks. Based on a classification that classifies user's motivation for checking in at a Point Of Interest into seven categories we propose an approach that will help refine recommendations in a way that can be better explained to the user. We also show the applicability of our approach by analyzing a dataset extracted from Foursquare.

CCS Concepts
• H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Information Filtering

Keywords
Location-based social networks; Point Of Interests Recommendations; Motivation-Aware; Explanations.

1. INTRODUCTION
Availability of multiple product choices and easy access to information about them has made the task of making correct purchase decision, by evaluating the information available, a huge problem for the consumers of the products or services. Recommender systems are software that helps customers make these decisions by providing them product recommendations that are relevant. Recommender systems give personalized recommendation to the user by either using explicit data provided by user through ratings or by using implicit data like user browsing behavior, past purchasing behavior etc. The popularity of personalized systems have increased manifold as today the success of e-commerce sites is dependent on the quality of recommendations. Hence, researchers are continuously trying to improve quality of recommendation by integrating more and more data about the customers in the recommendation process [1].

Presently, there is a clear trend towards usage of context-aware recommendation systems as they integrate contextual data like time, location, mood, emotions, companion, purpose etc. with ratings data to provide final recommendation[2]. Among the different contexts, research community has shown most interest towards location-aware recommendations systems. One reason for greater focus on location-aware recommendation systems is the easy availability of GPS data due to increased adoption of smart mobile phones.

Tourism industry is hugely impacted by the ubiquity of mobile phones in consumer lives [3]. Availability of many travel related apps and ease of access of free Wi-Fi spots has made mobile phones the main decision making tool in helping tourists make travel related decisions. Mobiles phones complemented with intelligent travel related apps has completely transformed the travel experience [4]. Among the technologies used for applications created for tourism, location aware and context aware based apps are the most popular as they have helped tourists to enhance their travel experience by making relevant recommendations. There is still a need for developing new approaches for recommending point of interests to tourists based on the variety of contextual and personal data available. This paper tries to address the above need by proposing a novel approach for recommending users places, restaurants, events etc. based on user motivation profile that is derived from his check-in data from location based social networks.

In this paper, we propose a novel approach for refining the recommendations made by location-aware recommender systems based on user motivations. Most existing recommender systems primarily use a combination of ratings data, content data like features or attributes of the product or service, and context data like location or time. We propose to integrate the user checking in motivation at places he has visited places into the location-aware recommendation system, as it will help refine recommendations in a way that can be better explained to the user. This will also lead to increased adoption of the recommendations as prior research has shown that explanation has been found more valuable by the user if they are explained in a more simple and accurate manner[1]. User motivation data is inferred from previous user check-in and comments at different locations. We also show how our approach can be applied through a case study on a real life dataset of 10 users extracted from popular location based recommendation app Foursquare.

2. RECOMMENDER SYSTEMS IN TOURISM
The key problems in recommender systems are the prediction problem and the top-N prediction problem[5]. The prediction problem is about predicting whether a user will like or dislike a new item that the user has not yet consumed or purchased. This prediction is generated using the knowledge of user preferences, past purchases data and interests. The top-N problem in recommender systems attempts to predict the set of N items that a user may like from the set of items he has not yet seen. Recommender systems in tourism industry primarily focuses on the top-N problem. In tourism industry these systems help the tourist or user in information search by recommending

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

destinations, point of interests, restaurants, events, travel itineraries etc. The recommendations made are specific to a user as they are personalized according to the user interests and preferences.

The popularity of recommender systems in tourism industry has brought this field into the attention of the academic research community. The increased focus on research in recommender systems in tourism is evident by going through the detailed and exhaustive survey papers [6], [7] that have been published on the topic recently. Among the recommendation problems that are researched in the tourism domain, Point of Interests recommendations (POI) is the most researched problem by the academic community [6].

In Point of Interests recommendations a ranked list of point of interests like tourists attractions in a city, restaurants, events etc. are presented to the user[8]–[11]. POI problem can be classified as top-N recommendation problem. These systems focus on two aspects of the problem, first on how to improve accuracy of the recommendations and the other aspect is how to effectively present the information to the user[8]. Majority of recommender systems in tourism focus on point of interests recommendations. One primary reason for that is the availability of new contextual data that has motivated researchers to focus on ways to improve recommendation accuracy. Location, time of the day, current weather, budget, means of transport, traffic, presence of friends nearby etc. [6]are contextual aspects that have been used in making POI recommendations. Location of the user is one the most popular contextual data that is used in most algorithms, one reason could be the easy access to accurate location data because of widespread use of mobile phones among tourists. Social network data is also used for making POI recommendations[10]–[12]. Social network data provides rich data points that can be used for profiling the user. It also provides data about relationship between users, preferences and views that can be derived from user comments, reviews and other network activity.

Tour Package [13] or Travel destination recommendation and Itinerary Planning [14], [15] are two more problems that have been researched. Travel destination recommendations are designed with tour operators as end users. These systems also recommend hotels, flights in addition to tourist locations. Cost is also one aspect that is considered an important criteria in tour recommendations[13]. Itinerary planning or route planning recommends multiple day personalized tour plans with set of point of interests to be explored each day. Contextual aspects like days of visit, pace of travel, preferred transportation mode [16] have been used for such recommendations.

Among the recommender systems approaches in tourism domain research, content based technique is the more popular as compared to collaborative filtering technique [6]. Unavailability of user rating data for different attractions, restaurants, events etc. may be the reason behind fewer collaborative filtering based approaches. Hybrid algorithms that combine content based and collaborative filtering based may be considered more appropriate for tourism domain recommendations.

3. RELATED WORK

Point of interest recommendations approaches in context based recommender systems is categorized by the type of data the systems process to make recommendations [17], [18]. Combining both the categorization approaches, POI recommendation approaches can be of six types.

Pure check-in data approach: This approach primarily considers check-in frequency data for making recommendations. It assumes that if two users are similar if they have similar checked in history. One demerit in considering check in data frequency as ratings is that during vacations tourists only check in once at a tourist location so it difficult to deduce whether the user liked or disliked the place.

Geographical influenced approach: The current location of the user and distance of POIs not yet visited by the user from the current location is used for making recommendations. This approach is appropriate when availability of time, transport options, traffic condition, weather conditions are used as contextual variables for making recommendation.

Social influence enhanced approach: Popularity of location based social networks like Foursquare, Yelp etc. have resulted in recommendation approaches that utilize social relationships among users to enhance POI recommendation. This approach assumes that friends of a user have similar interests as the user and a user is more likely expected to trust recommendations made by people who they are connected to in the network.

Temporal influence enhanced approach: Some POIs are preferred to be visited at a particular time slot, temporal influence approach considers time information while generating recommendations. For example, there are tourist locations that are primarily visited during sunrise or sunset time. Even closing time and opening time of museums and restaurants are important information that can help improve POIs recommendation.

Sequential influenced approach: These systems assume that users exhibits pattern in the order in which they visit places. For example, some users may prefer going to a restaurant after watching a movie or a game in a stadium. Patterns once identified from past check in data can be used for making recommendations.

Categorical influenced approach: Users preferences for checking in at particular categories of point of interests is leveraged in this approach. A user may prefer going to museums only and another user may have preferences for entertainment parks. The knowledge of a user biases for a particular category of POI is used in this approach for enhancing recommendations.

Among the different approaches for POIs recommendations, check-in data, geographical influenced and temporal influenced approach have significantly enhanced POI recommendation quality. Geographical influence is used the most to improve POI recommendation [17].

In [11] a approach is proposed that combines temporal and geographical data to make POI recommendation. Their approach splits time into hourly slots and mines the user checking in history to get insight about user temporal preferences to visit particular type of POIs at a time slot. As users tend to visit POIs that are closer to their current location, this approach combines the POIs nearby to user location with the insight acquired by the user temporal preferences to make the final recommendation.

Social network data, geographical data as well as check in data is used in the approach proposed in [19]. Their approach challenge the main assumption made in most POIs recommendations approaches that use location based social network (LBSN) data i.e. check-in frequency of user at a particular POI indicates user preference for that POI. This assumption is challenged on the basis that in more than 50 % of the places a user has checked in only once and on the basis of one check in it cannot be implied that the user prefers that POI. In the approach proposed by [19], they extract the preference of POI by mining user comments for that POI. The mining of the comments provide a sentiment polarity for the POI for that user. The sentiment polarity can be positive, negative or neutral. The final recommendation is made by integrating user sentiment polarity towards POIs he has commented on, user social network links and geographical location of the user.

Most approaches use geographical data, check-in data, and temporal data or combine them to make recommendations. An interesting approach [20] uses user personality data to enhance the recommendations. The personality of the user is captured through a questionnaire filled by the user during the registration process on the mobile application. The personality is based on the Five factor model [21]. The Five Factor Model terms personality among the five dimensions of Extraversion, Agreeableness, Conscientiousness, Neuroticism, and Openness to Experience. Along with personality of the user the approach uses a set of contextual factors, such as the weather conditions, the time of day, user's location and user's mood to recommend the final set of POIs.

Our approach uses the concept of user motivation for checking in as the context to refine the final recommendation. To the best of our knowledge, no other research paper has ever used this data for POIs recommendations.

4. MOTIVATION BASED RECOMMENDATION APPROACH

4.1 Motivation

Spatiotemporal mobility among user using location based social networks (LBSN) are driven primarily by social rewards and also by systems rewards [22]. Checking in behavior in LBSN is driven by users seeking status recognition in his network .LSBN enables social recognition as the feature of immediate sharing of location details, pictures etc. generates immediate social reaction among his network friends. Checking in behavior is an important aspect in marketing of services in LBSN. The authors cite the theory of self-concept [23] to explain the behavior of customers. Theory of self-concept indicates that consumers value consumption that results in recognition and that strengthen the conception about themselves. Similarly, we use motivation behind checking in at a location to refine recommendations as we believe that every user may have a different motivation behind checking in at a location. Using user motivation preferences while showing and explaining the final POIs recommendation to the users will result in more effective recommendations.

Our work is based on the foundation that users have a particular motivation when they check-in at a location. In this work, we use the classification done by [24], they found that motivations for a user to share his location or check-in at a particular location can be classified into seven categories.

They identify Social Enhancement, Informational Motive, Social Motivation, Entertainment value, Gameful Experience, Utilitarian motivation, Belongingness as the motives for a user to check-in at a location.

Social enhancement value is the most commonly observed motive, exhibited in more than fifty percent check-ins, where a user check-ins for impressing others and feels important to be at a place [25].

Information Motivation is commonly observed in youth, usually a suggestion or advice. Social Motivation is used when hanging out with friends or for relationship development.

Entertainment value is when user is relaxing or playing, to communicate positive moments.

Gameful experience is using gaming mechanics in non-gaming sense. City spots and achieving a virtual status like Mayor or owner. Utilitarian motivation is for winning promotions and discounts as you share or check-in at a place.

Belongingness is for places like home, school when users are nostalgic.

Scenario: Number of places a tourist can visit is limited because of the constraints of time and effort needed. POIs recommender systems help the users in deciding the POIs to visit using contextual variables. The final list to 2-3 POIs provided to the user as recommendation many times are difficult to justify as multiple contextual variables are evaluated using complex algorithms to generate the final recommendations. In our approach we further refine the final recommendations based on user motivation to checking in at a POI. The justification of the recommendations made through explanations based on user motivation for checking in will be easier for the user to comprehend.

For example, a tourist in Barcelona whose analysis of checking in data in Foursquare suggests that he is motivated by social enhancement will be recommended POIs like Sagrada Familia or Park Guell, while somebody who is motivated by information motivation will be recommended an offbeat attraction or a new restaurant.

4.2 Algorithm

Our aim is to recommend User U_i at location L_i a place of interest P_i that is within a radius of distance R_i from location L_i. We define two kind of motivations for each location or POI and for each user. The two motivations are Dominant explicit motivation and Dominant perceived motivation. Dominant explicit motivation for a user is derived from explicit data like comments and status messages after checking in at a POI on the location based social network. Dominant perceived motivation are generated for a location through survey.

We use the approach of explicit and perceived motivation because many users may not put any comments or status messages after checking in at a location. Using explicit motivation will more likely result in data sparsity.

Step 1: Assigning dominant explicit motivations to users and locations

Dominant explicit motivations for a user are determined based on the motivation inferred from the comments and status messages user have given after check in to different places. Set DUi represents the dominant motivations of a user U_i .It contains those motivations which have highest frequency of check-ins with a particular motivation. We have made DU_i a set as a user may have more than one motivation having the max frequency count. Similarly, Dominant explicit motivations to a place is referred as set DP_i and is determined by doing a frequency count of the inferred motivations derived from comments given to the place by users.

Step 2: Assigning dominant perceived motivations to users and locations.

Based on offline assessment of the places by a survey each place P_i is assigned a perceived motivation. PP_i is the set of dominant perceived motivations of a place P_i. It is determined by doing a frequency count of the perceived motivations assigned to the place P_i in the survey. PU_i is the set of dominant perceived motivations for a user U_i. It is determined by doing a frequency count of the perceived motivations assigned to each place the user U_i has checked into.

Step 3: Recommendation Generation
To recommend User U_i at location L_i a place of interest P_i that is within a radius of distance R_i from location L_i. Using collaborative filtering or other POIs recommendation algorithm approaches a set of places within a radius of distance R_i from location L_i are generated that are matching with user preferences based on his ratings or preferences data.

Step 4: Final set of motivation based recommendation
User U_i set of dominant motivations as generated in step 2 is the union of the sets DU_i and PU_i. Place P_i set of dominant motivations as generated in step 2 is the union of the sets DP_i and PP_i. Then the final set of recommendations is based on refining the places selected in step 3 using User U_i dominant motivations. From the set of places selected in step 3 only those places P_i whose dominant motivations matches with user U_i dominant motivations are recommended to the user U_i.

Our proposed algorithm approach applies post filtering contextual approach [2] as motivation context is applied on a list of recommendations generated by traditional recommender systems algorithms. A pre-filtering contextual approach can also be applied but as ratings data is primarily used by traditional algorithms, pre-filtering places of interest based on motivations may lead to data sparsity problem.

5. Case Study

Our approach as mentioned in the earlier section is to refine the recommendation made by an algorithm that is designed for accuracy. Our suggested approach objective is not to improve accuracy further but to improve the way final recommendations are explained to the user. Explanations[26] are an important component of recommender systems as it may increase the adaptability and trustworthiness of the recommender system. In [27], the authors show that there is merit in providing personalized explanations and explanation interfaces should be designed to increase the informativeness of the explanation. We believe our approach will add to the informativeness of the explanation.

Instead of an experimental evaluation of our approach we have done a data analysis on four square data set to check whether our approach is feasible in a real life scenario. Our approach is feasible only if users show variety of motivation while checking in, if all users show the same motivation then motivation cannot be used to refine the final recommendations. Our algorithm uses the concept of perceived and actual motivation, we also want to check through actual data whether there is any difference in actual and perceived motivation.

5.1 Data Collection

Foursquare launched in 2009 is used for check-in and real time location sharing with friends. It has 50 million users in its network and handles millions of check-in in a day. The Foursquare app allows the users to have their own profile and share their comment describing their feelings when they visit a location. The users of the foursquare were selected for the final analysis that has more than 10 check-in in Indore. We could find 10 users with such criteria who had visited in all 97 places including restaurants, pubs, city spots, home and business.

5.2 Comment Classification

The 7 motivations for check-in by [24] are used, Table 1 shows which characteristics of a comment can help us map with which motivation. For example, if a user checks-in at a high end restaurant and puts a comment "Tremendous food". Then his motivation would be classified as social enhancement value as it is a high end restaurant and the user has checked in as he is feeling important. Based on his comment the user motivation will be classified as information motivation. Similarly, all the comments by the user are classified by using characteristics of the motivation. Table 2 shows the result of classifying all the 129 comments made by the users in our dataset. The table shows the distribution of various motivations.

5.3 User Classification

Every user has one motivation from the above 7 categories. The motivation of the user is the highest frequency of motivation in the comments as classified according to the above method. Hence, a user *Ui* has a motivation *Mi*, if the comments posted by the user on foursquare has highest number of comments with *Mi* as motivation. In our dataset of 10 four square users in Indore, 50 per cent had Social Enhancement value as their main motivation. What was surprising was that both social enhancement and Informational motivation together were dominant motivation in 20 per cent user. Hence, for a user it is not necessary to have a single motivation as a dominant motivation but combination of more than one. Table 3 shows classification of users on the basis of 7 motivations.

Table 1. Characteristics of different motivations for location check-ins

Motivation	Characteristics
Social Enhancement	Impressing others Feeling Important To show off Extremely Popular location Night clubs High end restaurants Distinctive Identity or Intellectual Image Celebrity Status
Informational Motivation	Suggestions Advices Information about event or news Location and arrival Important event Give and take recommendation
Social Motivation	Meeting new people Socializing Observing others Meeting a Friend Flirting and relationships Emotional Feeling At Home or Office Know about friends and where they are
Entertainment Value	Playing Relaxing

	Passing Time Less Lonely Positive moments, emotional state Fight boredom Initiate chat Waiting
Gameful Experience	To collect award Points Status in an app More in females Older age(not in youth aged 19-22) City spots(streets, square, roads, bridge, old town) Escape from reality, Virtual Possession
Utilitarian	Win promotions and discounts Check in of family business for marketing
Belongingness	Place with social group Nostalgia or ownership

Table 2. Comments of Four Square dataset classified on motivation for check-in

Motivation	Per cent of comments
Social Enhancement Value	38
Informational Motivation	27
Social Motivation	7.7
Entertainment Value	10.0
Gameful experiences	3.8
Utilitarian Motivation	3.8
Belongingness	9.30

Table 3. Spread of Motivation among users in our dataset

Motivation	Per-cent
Social Enhancement Value	50
Informational Motivation	10
Social Enhancement Value and Informational	20
Social Motivation	10
Belongingness	10

5.4 Perceived & Actual Motivation

While the user giving a comment on a location he visits might be classified into one of the motivation category, but this motivation may differ for the perceived dominant motivation of the location. The perceived dominant motivation of the location is classified based on a survey. This mismatch in perceived and actual motivation in check-in can lead to distorted image of the user. For example, suppose a user checks-in at a high end posh restaurant with a comment "Excellent coffee, Must try." Though, the actual motivation of the user is Information Motivation but the characteristics of the place may make another user who sees this comment assume the motivation behind check-in was Social Enhancement Value. To address this dissonance, in step 4 of the algorithm, for a User U_i, the set of dominant motivations is generated by the union of the sets DU_i and PU_i. We analyzed the data to check whether this kind of dissonance exists in our data set. Table 4 shows 39% of times the actual motivation is also the perceived motivation but a majority number of times the perceived and actual motivation differs. Also, 12 percent places had multiple classifications which dint allow us to attach them to a specific motivation.

Table 4. Difference between actual and perceived motivation

Perceived & Actual Motivation	Places	Percentage
Equal	35	39.32584
Not Equal	43	48.31461
Not Determined	11	12.35955

6. DISCUSSION & CONCLUSION

Every user has a motivation when the user checks-in at a particular location, if these motivation is taken into account while generating final recommendations then it will be more beneficial to the user. Context variables like time, location and social network data of a user are mainly used to recommend new locations to a tourist. In this paper, we propose an approach that uses user checking in motivation along with the other contextual variables. Motivation can be effectively used if used as a post-filtering contextual variable in combination with the existing recommendation algorithm. Our analysis of real life data shows that our approach can be used as user do show different motivations as they check in into different POIs , the primary motivation among users also differs and there do exist a difference between a user's actual motivation for checking in and perceived motivation for checking in. We believe using our approach will improve the explanation quality of the final recommendations.

Limitations of the study are that we did not experimentally evaluate the accuracy of our approach based on metrics like mean absolute error, precision or recall. Our approach is not designed to improve accuracy, what it offers is the additional explanation for a recommendation to the user, which helps him understand the recommendation given more easily. In future research we aim to operationalize this algorithm on a mobile app and then try to do a qualitative evaluation of the ability of the algorithm in providing more satisfaction to the user. Though, the existing dataset is sufficient for gaining insights on the appropriateness of our algorithm, a qualitative study is required to show the benefit of using checking in motivation for enhancing POIs recommendations.

7. REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, 'Recommender systems survey', *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[2] G. Adomavicius and A. Tuzhilin, 'Context-Aware Recommender Systems', in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 217–253.

[3] J. E. Dickinson, K. Ghali, T. Cherrett, C. Speed, N. Davies, and S. Norgate, 'Tourism and the smartphone app: capabilities, emerging practice and scope in the travel domain', *Curr. Issues Tour.*, vol. 17, no. 1, pp. 84–101, Jan. 2014.

[4] I. P. Tussyadiah, 'The Influence of Innovativeness on On-Site Smartphone Use Among American Travelers: Implications for Context-Based Push Marketing', *J. Travel Tour. Mark.*, vol. 33, no. 6, pp. 806–823, Jul. 2016.

[5] M. Deshpande and G. Karypis, 'Item-based top- *N* recommendation algorithms', *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, Jan. 2004.

[6] J. Borràs, A. Moreno, and A. Valls, 'Intelligent tourism recommender systems: A survey', *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7370–7389, Nov. 2014.

[7] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, 'Mobile recommender systems in tourism', *J. Netw. Comput. Appl.*, vol. 39, pp. 319–333, Mar. 2014.

[8] J. M. Noguera, M. J. Barranco, R. J. Segura, and L. Martínez, 'A mobile 3D-GIS hybrid recommender system for tourism', *Inf. Sci.*, vol. 215, pp. 37–52, Dec. 2012.

[9] F. Ricci, 'Travel recommender systems', *IEEE Intell. Syst.*, 2002.

[10] J. J.-C. Ying, E. H.-C. Lu, W.-N. Kuo, and V. S. Tseng, 'Urban point-of-interest recommendation by mining user check-in behaviors', 2012, p. 63.

[11] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M.- Thalmann, 'Time-aware point-of-interest recommendation', 2013, p. 363.

[12] N. Saiph Savage, M. Baranski, N. Elva Chavez, and T. Höllerer, 'I'm feeling LoCo: A Location Based Context Aware Recommendation System', in *Advances in Location-Based Services*, G. Gartner and F. Ortag, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 37–54.

[13] Y. Ge, H. Xiong, A. Tuzhilin, and Q. Liu, 'Cost-Aware Collaborative Filtering for Travel Tour Recommendations', *ACM Trans. Inf. Syst.*, vol. 32, no. 1, pp. 1–31, Jan. 2014.

[14] Y. Kurata and T. Hara, 'CT-Planner4: Toward a More User-Friendly Interactive Day-Tour Planner', in *Information and Communication Technologies in Tourism 2014*, Z. Xiang and I. Tussyadiah, Eds. Cham: Springer International Publishing, 2013, pp. 73–86.

[15] R. Schaller and D. Elsweiler, 'Itinerary recommenders: how do users customize their routes and what can we learn from them?', 2014, pp. 185–194.

[16] A. Garcia, P. Vansteenwegen, O. Arbelaitz, W. Souffriau, and M. T. Linaza, 'Integrating public transportation in personalised electronic tourist guides', *Comput. Oper. Res.*, vol. 40, no. 3, pp. 758–774, Mar. 2013.

[17] Y. Yu and X. Chen, 'A survey of point-of-interest recommendation in location-based social networks', presented at the Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[18] J.-D. Zhang and C.-Y. Chow, 'Point-of-interest recommendations in location-based social networks', *SIGSPATIAL Spec.*, vol. 7, no. 3, pp. 26–33, Jan. 2016.

[19] J.-D. Zhang, C.-Y. Chow, and Y. Zheng, 'ORec: An Opinion-Based Point-of-Interest Recommendation Framework', presented at the Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1641–1650.

[20] M. Braunhofer, M. Elahi, and F. Ricci, 'Usability Assessment of a Context-Aware and Personality-Based Mobile Recommender System', presented at the International Conference on Electronic Commerce and Web Technologies, 2014, pp. 77–88.

[21] R. R. McCrae and O. P. John, 'An Introduction to the Five-Factor Model and Its Applications', *J. Pers.*, vol. 60, no. 2, pp. 175–215, Jun. 1992.

[22] I. P. Tussyadiah, 'A Concept of Location-Based Social Network Marketing', *J. Travel Tour. Mark.*, vol. 29, no. 3, pp. 205–220, Apr. 2012.

[23] E. L. Grubb and H. L. Grathwohl, 'Consumer Self-Concept, Symbolism and Market Behavior: A Theoretical Approach', *J. Mark.*, vol. 31, no. 4, pp. 22–27, 1967.

[24] R. Yavuz and A. Toker, 'Location sharing on social networks: implications for marketing', *Mark. Intell. Plan.*, vol. 32, no. 5, pp. 567–585, Aug. 2014.

[25] U. M. Dholakia, R. P. Bagozzi, and L. K. Pearo, 'A social influence model of consumer participation in network- and small-group-based virtual communities', *Int. J. Res. Mark.*, vol. 21, no. 3, pp. 241–263, Sep. 2004.

[26] N. Tintarev, 'Explanations of recommendations', 2007, p. 203.

[27] A. Sharma and D. Cosley, 'Do social explanations work?: studying and modeling the effects of social explanations in recommender systems', 2013, pp. 1133–1144.

Research Methods for Group Recommender Systems

Amra Delic
E-Commerce Group
TU Wien
Vienna, Austria
amra.delic@tuwien.ac.at

Julia Neidhardt
E-Commerce Group
TU Wien
Vienna, Austria
julia.neidhardt@tuwien.ac.at

Thuy Ngoc Nguyen
Free University of Bozen-Bolzano
Bolzano, Italy
ngoc.nguyen@unibz.it

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

ABSTRACT

In this article we argue that the research on group recommender systems must look more carefully at group dynamics in decision making in order to produce technologies that will be truly beneficial for users. Hence, we illustrate a user study method aimed at observing and measuring the evolution of user preferences and actions in a tourism decision making task: finding a destination to visit. We discuss the benefits and caveats of such an observational study method and we present the implications that the derived data and findings may have on the design of interactive group recommender systems.

CCS Concepts

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

Keywords

Group Decision Making, Group recommender systems, Observational Study

1. INTRODUCTION

Recommender systems for groups are becoming more and more important since many information needs originate by group and social activities, like listening to music, watching movies, traveling, attending sport events, and many more. The importance of group recommender systems also has increased due to the social web, where users are not isolated but form interrelated groups. A high number of papers on group recommender systems have been published [13] but still, we believe, there is a gap between the current main focus of the research and the information search and decision making support needs of groups.

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

Research on group recommender systems often focuses on aggregation strategies, i.e., how to combine individual preferences, sometimes conflicting preferences, into a group profile. According to Arrow's theorem, it is clear that an optimal aggregation strategy does not exist - group recommender systems studies also confirmed that there is no ultimate winner. There are only a few studies that concentrate on decision/negotiation support in group recommender systems: *Travel Decision Forum* [12], *Trip@dvice* [2], *Collaborative Advisory Travel System* (CATS) [14], *Choicla* [21]. To our best knowledge, there are no observational studies on group decision processes in the context of group recommender systems. These types of studies are usually conducted in the social disciplines: in [22] the importance of discussions, especially with respect to information that is shared among group members is emphasized. An extensive overview of studies on group dynamics and the influence of the different aspects (e.g., group structure, group decision process structure) on the group choices is presented in [8].

The main motivation of this paper is therefore to raise in the group recommender systems community the awareness of the importance of a new type of analysis: observing groups in naturalistic settings. We believe that the design of a novel and more effective sort of group recommender systems can be initiated if one better observes and understands groups in actions, measures their behaviors, and tries to identify concrete opportunities for computerized systems to become more useful to people. In this paper we will illustrate the design, the outcome and the implications of an observational study where groups of people faced a concrete decision task - select a destination to visit as a group - and the researchers monitored the groups before, during and after the task.

Hence, our study is motivated by a range of dimensions and issues, that we list in the following.

- *Decision making* is the ultimate motivation for a group recommender system. This is true even more than for individual recommenders which can also be used for expanding user knowledge or expressing self [20]. But if group recommenders must support decision making we must understand how this task is executed in groups and how the decision issues, the group members and the contextual situation alltogether impact on it. In the past too much attention was put on how to identify "optimal" recommendations, which in the context of groups is not even possible to correctly define.

- We believe that the *application domain* is crucial in a group recommender system. Recommending tourist attractions or destination for a group cannot follow the same model used to recommending movies to watch [24]. The tourism product is more complex than other types of products (i.e., it is a bundle of products and services) and in the same time it is less tangible. Moreover, traveling is an emotional experience and explicit preference characterization is problematic especially in the early phase of the travel decision-making process as different users usually have different perceptions of the features of the items. Finally, tourism products are typically experienced in groups. For that reason, we have tried to generate a decision task - destination selection - that is believable in the context of tourism decision making and we made observations for users characteristics and decision outcome that have emerged as important in tourism research on consumer behavior [6, 7, 23, 25].

- *Group recommendations techniques* have been influenced too strongly by social choice theory [13] and not enough by group dynamics studies [8]. It is still unclear how a recommender can identify items to suggest in a group decision making task, if the goal is not simply to aggregate the votes/preferences expressed by the group members. But we believe that studies like the presented one can help to understand the key information that groups need in order to make decisions, which could not simply be the suggested outcome of the decision. We believe that the more general concept of information recommendation, rather than product recommendation, is important to implement [3].

- It is clear to us that the design of more effective group recommender systems requires a *multidisciplinary approach*. In that sense the study described in this paper brings together social science and computer science scholars. Observational studies are not part of the classical research repertoire of recommender systems research methods, but, we believe that these methods are strictly required if we want to understand users in naturalistic settings and be able to generate fruitful conjectures about new and useful system functions.

- Another important motivation of this study is the desire to *collect data* about group decision making that can be exploited by several research groups. Hence, in some sense, we wanted to obtain raw data that could be used to several types of analyses, from different perspectives and with alternative motivations. We plan to make the data that we have collected, and that will also be collected in future implementations of the study, available to everyone for further analyses.

- Finally, we believe that the research community on group recommender systems needs to discuss and build a *research agenda*. We must identify critical challenges and expected results. In this study we initiate this reflections by raising several issues, e.g., how to measure the collective behavior of a group, what properties of a group are more important in recommender systems and how they should be measured, how to define group satisfaction, how to compare and relate user preferences and group preferences.

Thus, the aim of this paper is to reflect on research methods for group recommender systems on the basis of an observational study.To present a detailed analysis of the collected data is not the focus of this paper; this was done in [5].

The rest of this paper is structured as follow: in Section 2 the study procedure is described in detail, Section 3 illustrates instruments used for the data collection, in Section 4 results of a first analysis are summarized, followed by Section 5 where implications for recommender systems are explained. Finally, in Section 6 we discuss limitations, challenges and possible variations of the study.

2. PROCEDURE

In order to design a new generation of more useful and effective group recommender systems, we do not only aim at gaining insights into human behavior, but also at learning how to improve and facilitate interaction of users in a computer mediated setting. To set a basis for this, we started with an exploratory research approach that is not constrained by any pre-existing system functionality, i.e., we developed a study to collect observational data on human-to-human interactions in group decision making task. In the following we describe the procedure of this observational study in detail.

The study was initiated in a cooperation with the International Federation for Information Technologies in Travel and Tourism (IFITT) and 11 universities worldwide. The first implementations of the study took place at the Delft University of Technology (TU Delft), the University of Klagenfurt (UNI Klagenfurt) and the University of Leiden (UNI Leiden), while an extended study was carried out at the Vienna University of Technology (TU Wien). Each implementation was conducted as a part of a regular lecture and followed a three-phases structure: pre-survey questionnaire phase, groups meeting/discussion phase and post-survey questionnaire phase (see Figure 2).

Prior to the first study phase, an introduction with general instructions for the participants was presented. The first task for all participant was to form groups. At TU Delft, UNI Klagenfurt and UNI Leiden, students were free to choose their group size (between two and four group members). At TU Wien students were instructed to form groups of six members and to select two students (referred to as *observers*) whose task was to observe and record activities of their group in the next phase. All the other group members took part in the decision making process (referred to as *decision makers*).

In the first study phase, the task for the decision makers was to fill in a pre-survey online questionnaire that captures their individual profiles, preferences and dislikes. Detailed data description is provided in section 3. Also, in this phase, in Vienna, a short training for observers was organized. The purpose was to introduce them with the following study tasks and to instruct them on how to perform and record a group observation. A report template, which was constructed based on Bales's Interaction Process Analysis (IPA) [1], was provided to the observers to record the activities of the decision makers. The observers also received written instructions and during the rest of the study they were in a close contact with the study organizers.

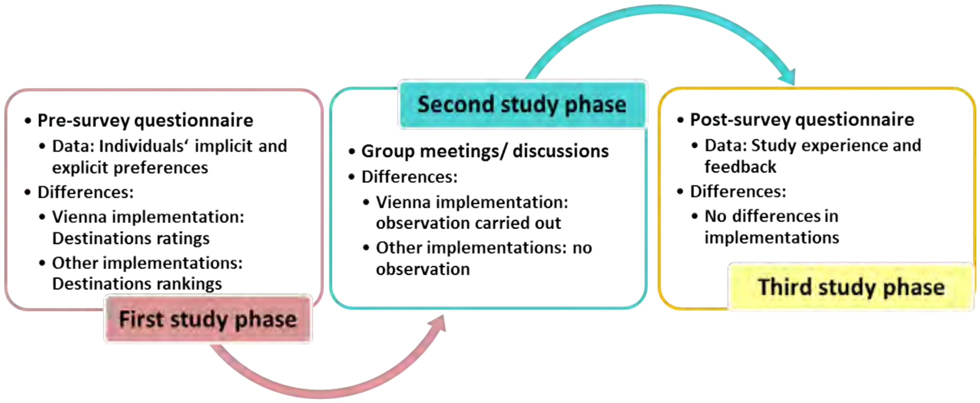


Figure 1: Overall structure of the study and differences between implementations

In the second study phase, the group meeting and discussion took place. The decision makers received written instructions with the following structure:

1. Ten predefined destinations together with informational Wiki pages;

2. Decision task scenario: *Imagine that you are working on a research paper together with the other group members. Interestingly, your university offers you the opportunity to submit this paper to a conference in Europe. If the paper gets accepted, the university will pay to each group member the trip to the conference. In addition, you will be able to spend the weekend after the conference at the conference destination. Ten conferences will take place in European capitals around the same summer period*;

3. Next, they were asked to discuss and decide which destination they would like to visit most as a group. Additionally, they also had to provide a second choice in case that the first option would no longer be available.

Groups were not instructed on how to perform the decision making task and whether they should check the informational Wiki pages or not. This specific design was chosen due to its simplicity. Usually, when a group is planning a trip a bundle of different trip aspects have to be considered, e.g., timing, budget, destination, accommodation, transport, etc. This type of task would be almost impossible to simulate in a controlled environment. Thus, we concentrated on a simple aspect to analyze the basis of group interactions and dynamics in this specific context. At TU Wien, observers were included in the task. They audio recorded and reported the group decision process using the previously mentioned report template (details in 3).

In the third phase, the decision makers filled in an online post-survey questionnaire inquiring about the previous phase and the overall experience. During this phase, interviews with the observers were arranged in Vienna: for each group a meeting with the two observers of that group took place. Firstly, we evaluated observers' understanding of the task and the reports that they submitted, then, the observers elaborated their reports and discussed differences between those. Furthermore, they were also queried about the behavior of the decision makers and how seriously they actually performed the task.

At each university the study implementation followed the described structure. However, still some differences existed, they are explained in section 6. After the first implementation round, considering all the locations where the study was conducted, the size of the collected data sample comprised 78 decision makers in all together 24 groups of two, three and four group members, plus 16 observers (two for each group) at TU Wien. At TU Delft, after a first implementation round (referred to as *TU Delft*), a second one with the same configuration (without observation) took place (referred to as *TU Delft2*). It introduced 122 new decision makers in 31 groups. Thus, currently the data sample comprises 200 decision makers in 55 groups of two, three, four and even five group members (see Table 1) plus 16 observers.

Group size	**2**	**3**	**4**	**5**
UNI Leiden	2	2	2	/
UNI Klagenfurt	1	1	4	/
TU Delft	1	2	1	/
TU Delft2	1	8	14	8
TU Wien	2	1	5	/
SUM	**7**	**14**	**26**	**8**

Table 1: Groups sizes per university

3. MEASUREMENTS

In this section we describe the data in detail as well as the instruments were used to collect it: a pre-survey questionnaire, a template for reporting the observations and a post-survey questionnaire. Each of these instruments was designed in a way that the obtained data cover different aspects, which might impact the group decision process and which were derived from the literature.

Accordingly, the first data collection instrument - a pre-survey questionnaire[1] captured individual profiles of the participants in a similar way as the user profile in a recom-

[1]https://survey.aau.at/2012/index.php?sid=49577&lang=en

mender system would be represented. It is comprised of 68 questionnaire statements separated into four sections:

1. Demographic data and university affiliation (i.e., age, gender, country of origin, university and student identification number);

2. 17 tourist roles and Big Five Factors:

   - 30 questionnaire statements related to 17 tourist roles (i.e., types of touristic short term behavior) defined in [10];
   - 20 questionnaire statements related to the Big Five Personality Factors (i.e., Openness to new experiences, Conscientiousness, Agreeableness, Extroversion, Neuroticism) [11].

3. Experience and ratings/ rankings of ten predefined destinations:

   - Destinations: Amsterdam (at TU Wien and UNI Klagenfurt), Berlin, Copenhagen, Helsinki, Lisbon, London, Madrid, Paris, Rome, Stockholm and Vienna (at TU Delft and UNI Leiden);
   - Participants were asked how many times they have visited each destination;
   - Participants at the TU Wien rated, while other participants ranked the ten destinations (implications of this distinction are discussed in section 6).

4. Ranking of decision criteria (i.e., budget, weather, distance, social activities, sightseeing and other).

A five-point likert scale was used for the 50 questionnaire statements related to the 17 tourist roles and the Big Five Factors. To obtain the scores, i.e., the level to which a person belongs to a certain tourist role or to a certain personality trait, ratings of the statements were summed and divided by the number of related questionnaire statements. Tourist roles and personality traits are related to the user model of the picture-based recommendation engine (see section 5).

In the second phase group decision task took place. By now, only at the TU Wien, observational part of the study was implemented. The report template for the observers' recordings was designed based on the Bales's Interaction Process Analysis (IPA) (i.e., a method to study small groups and interactions among group members) [8]. Thus, the task for observers was to audio record group discussion and to fill in the provided report template. The report template consisted of the following sections:

1. Whether a plan for the group decision process was used or not and if yes the duration of the different decision process phases. We note that in [8] a four phases structure for the decision making process is indicated as typical: 1) Orientation, 2) Discussion, 3) Decision and 4) Implementation and evaluation of the decision;

2. Group members' roles (e.g., leader, follower, initiator, information giver, opinion seeker);

3. Group members' behavior (i.e., twelve categories of behavior: Show solidarity/ "Friendly"; Show tension release; Agree, Give suggestion/ opinion/ information; Ask for suggestion/ opinion/ information; Disagree; Show tension) - For each group member, the observers were requested to identify, record and categorize each "unit" of interaction (i.e., verbal and non verbal expressions) according to the twelve categories of behavior;

4. Social decision scheme (i.e., delegating, averaging, voting, reaching consensus or other -explanation could be provided);

5. Strength of group members' preferences (i.e., for each group member, the observers rated from *1 - Very unwilling* to *5 - Very willing* on how willing they were to give up on their preferred options).

Finally, a post-survey questionnaire[2] was used to collect data about the participants' experience with the group decision process and the overall study. It asked for:

1. The first and the second group choice;

2. Whether the provided information about the destinations was used during the group decision process;

3. Description of the decision process that led the group to their final choice;

4. Overall attractiveness of the ten predefined destinations (e.g., *"Many destinations were appealing."*, *"I did not like any of the destinations."*);

5. Satisfaction with the group choice (e.g., *"I like the destination that we have chosen"*);

6. Difficulty of the decision process (e.g., *"Eventually I was in doubt between some destinations."*);

7. Participant's perceived identification and similarity with the other group members (e.g., *"I see myself as a member of this group"*, etc.);

8. Assessment of the task (i.e., participants were asked to select the statements to which they agree regarding the organization of the task, their feedback and willingness to participate in the same or similar study).

A five-point likert scale was used to assess 4., 5., 6. and 7.

The overall structure of the data is shown in Figure 2. It visualizes the data as an Entity Relationship Diagram (ERD). Different colors indicate different study phases, i.e., pink: pre-survey questionnaire, blue: groups meetings/ discussions and yellow: post-survey questionnaire. Central entity in the ERD is the group member, i.e., the decision maker who is connected to all the other data dimensions (for the observers, only the demographic data is collected).

4. THE OUTPUT

In this section we summarize some concrete output obtained by an initial analysis of the data [5]. However, this is only one example how this type of studies can help to obtain deeper insights into the interplay of individual preferences and group processes. Various other analyses can be conducted making use of the rich information that has been (see Section 3). To facilitate this, we plan to provide the data to the research community.

[2]https://survey.aau.at/2012/index.php?sid=98597&lang=de

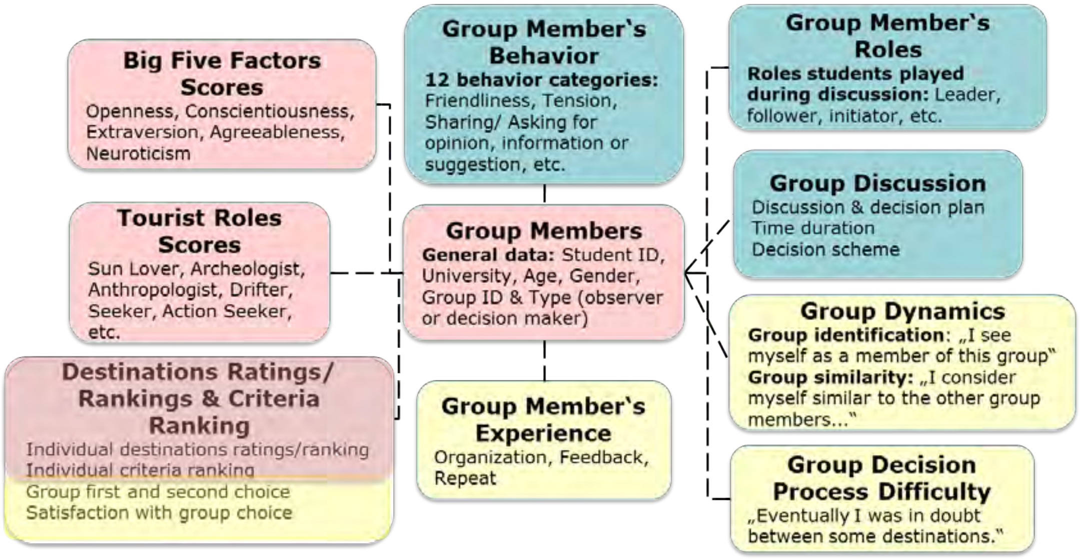


Figure 2: Structure of the collected data

(**Pink** - First study phase; **Blue** - Second study phase; **Yellow** - Third study phase)

In a first step, we studied whether or not the users were satisfied with the outcome of the group decision making process, and we tried to understand the impact of the initial preferences into that. The vast majority of users showed a high satisfaction for the destination chosen by the group. Obviously this was particularly true for users, where the group selection matched their individual top choice. However, also more than two-thirds of the users, for whom the group decision was not in accordance with their most preferred destination, were satisfied with the collective choice. To some extent this might be related to the fact that the users perceived the different destinations, which could be chosen for the group tour, overall as very attractive. However, our analysis clearly indicated that the group decision making process itself played a decisive role in this context: group preferences are not just an aggregation of the initial group members' preferences but are rather constructed during the process. This was also supported by the fact that common aggregation strategies in group recommender systems were hardly able to predict the outcome of the group decision making process.

Next, we studied the choice satisfaction of the users in more detail and identified relevant user and group characteristics in this context. We found some significant and moderately high correlations between the individual choice satisfaction and personality traits of a user. Also behavioral patterns during the discussion could be related to the satisfaction of a user as well as the difficulty of the task. To capture the satisfaction of a group, we studied the average choice satisfaction of the group members. Statistical tests identified significant differences between highly and less satisfied groups with respect to a number of factors. These factors captured, on the one hand, whether or not the group perceived the task as difficult. On the other hand, they were related to aggregated travel behavioral patterns as well as personality traits of the group members. Furthermore, in less satisfied group typically all members show disagreement during the decision making process.

5. IMPLICATIONS FOR RECOMMENDER SYSTEMS

As mentioned previously, the proposed observational study is ultimately motivated by the goal of designing more effective group recommender systems. This means that the system should better predict, and therefore recommend, which items the group will choose and will make the group members more satisfied. We will now discuss some important benefits that we expect the analysis of the data acquired by observing users' interactions in group decision making tasks can bring to recommender systems.

First of all, group recommenders requires the design of ranking functions that can highlight which items a group must primarily look at. Ranking functions for group recommender are based on preference aggregation strategies. While we already mentioned that there is not a single best aggregation strategy that fits all recommendation tasks and decision contexts, observational study data can be used to choose and customize the aggregation function to the specific contextual conditions of the group. We conjecture that, having a family of candidate aggregation functions, one can optimally choose the right one by fitting the observation data. For instance, experimental results of the study showed that the social role and personality of the group members influence group choices which was also confirmed in other studies [9], [18], [19]. Hence, for instance, among a family of multiplicative aggregation models one can fit the importance weights of the group members depending on their roles and personality.

A second important usage of observational data is the construction of a more dynamic model of recommendation that integrate into the baseline user preference models preference information derived by the observations of the discussion

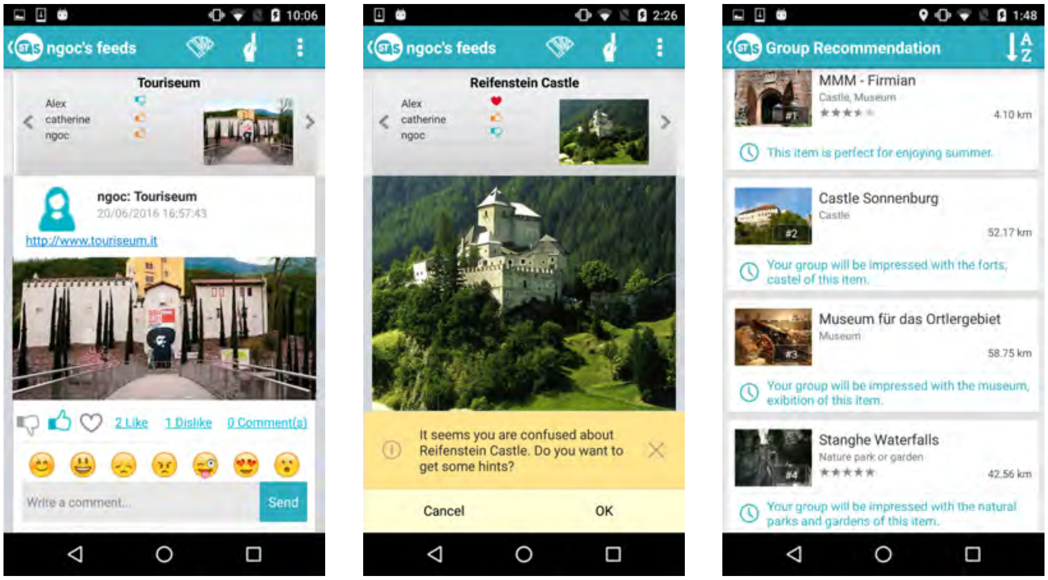


Figure 3: Screen-shots of STSGroup, from left to right: (a) Group discussion, (b) Hint suggestions, (c) Group suggestions.

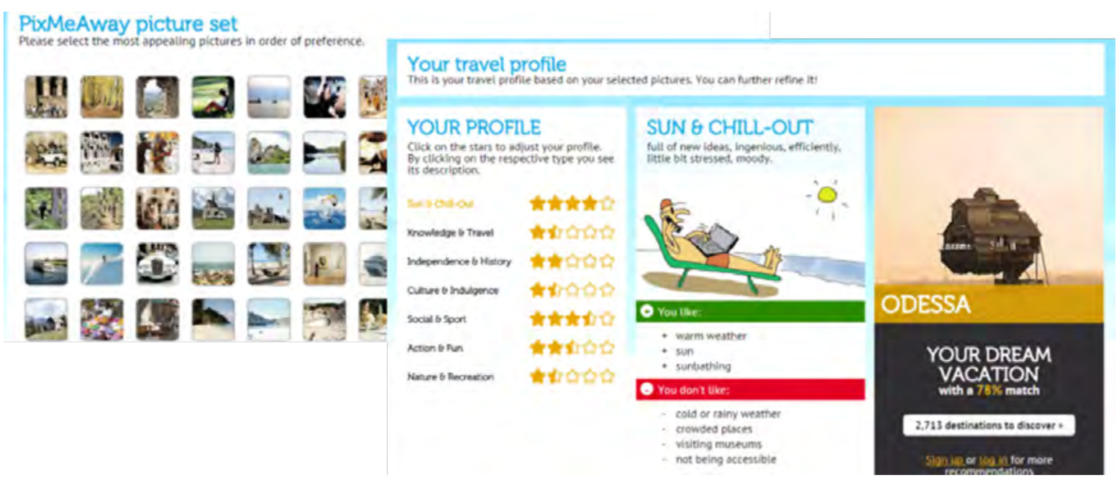


Figure 4: Screen-shots of the picture-based recommendation engine PixMeAway

process. In fact, it is clear from our study that the final output decision is not completely determined by the initial preferences of the users. We conjecture that the observed dynamic of the users-to-users interactions must be considered in order to better predict which items may suit the group at that precise point in time. We have for instance mentioned the observed correlation between the user activity in providing information or criticizing options and the satisfaction for the final choice. As we suggested in the paragraph above, also this data can be used to identify a better aggregation function. But, we also conjecture that this type of information can be exploited to revise the initial user models learned by the system using the historical preference data of the users. For instance, if a content based model was fitted to the known ratings of a user, this model can then be revised by considering the items that the user liked or criticized. An initial prototype implementing this idea is presented in [17]. That mobile system, which is called STSGroup, allows group members to be engaged in a discussion where they can propose items that are thought to be suitable for their group and react to other group members' proposals by giving feedback such as likes, dislikes or favorites. They can also tag the proposed items with comments and emoticons as shown in Figure 3a. The interactions between the members and the system during the group discussion are monitored and taken into account in order to actively provide group members with appropriate directions and recommendations (see Figure 3b and Figure 3c). The group recommendations are built up with explanations that are computed on the base of the group members' actions and contexts.

A third, probably most fundamental issue, is related to the ultimate goals of observational data and the scope of a group recommender system. Should the recommender fit the data, i.e., suggest what the users in a given context are supposed to choose, or should instead the system act as a mediator, aimed at driving the group towards a more fair choice? In the first case, as illustrated in the two paragraphs above, the system pleases the group and let it more smoothly and efficiently converge towards the decision that the group may have taken even without the system intervention. In the second case, the system is instead assuming that the fairness of a sound aggregation strategy should prevail on the natural group dynamics and will stick to it. This contraposition is not new in recommender systems: it relates to the question whether a recommender should only suggest items predicted to be top choices for the user or inject in the recommendations items that would make the list of recommendations more diverse, novel, sustainable, or simply more trendy. In order to address these fundamental questions, and understand which role the recommender should play, live user studies are unavoidable.

A fourth, very concrete implication of the study is related to the picture-based approach introduced in [15, 16]. The pre-survey questionnaire and the picture-based approach lean upon the same dimensions when capturing a user model, i.e., 17 tourist roles and the Big Five Factors. The findings of the observational study will be related to the picture-based approach model, which is illustrated in Figure 4, and then generalized to a group recommender system. The proposed research and related challenges are described in [4].

6. DISCUSSION

In this section we summarize the contributions of the paper and mention several challenges that have to be addressed when analyzing the data. Furthermore, we discuss potential variations and generalizations of the observational study.

The main contributions of the paper are:

- A detailed description of the replicable study procedure and the instruments used for the data collection that can provide insights into the actual group decision making processes.
- The implementation of the study procedure in a concrete context of tourism and traveling.
- Experimental results showing that certain individual and group characteristics, which go beyond the initial preferences of the individuals and their straightforward aggregation, play an important role in the final choice of the group.
- The implications of the observational study for group recommender systems and different aspect that should be considered when building such systems.

During the initial data analysis, we encountered several challenges related to data measurements we used. These challenges are at the same time limitations of the study and need to be addressed in the future work:

1. How to aggregate different individual scores, e.g., personality traits, at the group level?
2. How to measure diversity among group members with respect to the different data dimensions?
3. How to distinguish satisfied from not so satisfied groups?
4. How to match and compare individual preferences to the preferences of the group as a whole?
5. How to address ratings/ ranking difference in different study implementations?
6. How to relate participants' personalities to their preferences?

So far, we were mainly using the average of the individual scores when aggregating them at the group level [5]. However, more sophisticated approaches will be applied in future work.

Different dimensions of the study procedure can be varied in order to grasp diverse insights into the group dynamics in this particular context. In the following we present some of the variations and their potential implications:

1. *Duration and timing of the study*: In our implementations, we noticed different behaviors of the students in the study conducted over the three weeks period on the one hand and the study conducted in one lecture session on the other hand. In the first case students were not explicitly referring to their initial, individual preferences, but were rather discussing their preferences in general. In the second case, students were comparing their initial preferences and their final choice was based on these comparisons.
2. *Diversity of the ten predefined destinations* (e.g., country side tourism vs. big city tourism; mountain destination vs. sea side destination; hot climate destination vs. cold climate destination): Higher diversity could generate more conflicting preferences in groups and more intense discussions and decision processes.
3. *Locality of the ten predefined destinations*: In our case the ten destinations (but Amsterdam) were capitals in Europe and in an hour or two flight distance. By changing the locality of the chosen destinations would there be some differences in the observed decision process? Furthermore, the locality and overall popularity of the ten chosen destinations were related to the knowledge that the participants possessed about these destinations. But, by using less known destinations, how would the unfamiliarity with the destinations influence the decision process?
4. *Groups size*: The conducted data analysis showed differences in groups' satisfaction with respect to the group size - smaller groups tend to be more satisfied with the group choice than the larger groups, which is quite intuitive. Nevertheless, varying the group size in the study can provide insights in different aspects that should be considered.
5. *Budget*: Including budget into the group discussion increases the complexity of the task for the participants and it also enables more realistic setting of the decision process in the context of traveling.
6. *Group decision task*: If the group were to choose a point of interest that they *actually* had to visit together right after the group discussion, then the group members might pursue their preferences and interests in a more natural manner and more persistently.

7. *Domain*: The same study could be carried out in a different domain, such as music, movies, restaurant, etc. In this case it would be much easier to introduce a realistic setting to participants, but the discussion process, in this case, would clearly be much different.

To summarize, in this paper we presented the observational study implemented at several universities, the instruments used for the data collection and described the collected data. We stressed the implications of the study for group recommender systems and our future work relying on the founding of this study. At the end, we outlined main contributions, introduced challenges and limitations detected by now.

7. REFERENCES

[1] R. F. Bales. A set of categories for the analysis of small group interaction. *American Sociological Review*, 15:257–263, 1950.

[2] P. Bekkerman, S. Kraus, and F. Ricci. Applying cooperative negotiation methodology to group recommendation problem. In *Proceedings of Workshop on Recommender Systems in 17th European Conference on Artificial Intelligence (ECAI'06)*, pages 72–75, 2006.

[3] H. Blanco and F. Ricci. Inferring user utility for query revision recommendation. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18-22, 2013*, pages 245–252, 2013.

[4] A. Delic. Picture-based approach to group recommender systems in e-tourism domain. In *Conference Proceedings of the 24th Conference on User Modeling, Adaptation, and Personalization (UMAP 2016)*, Halifax, Canada, 2016.

[5] A. Delic, J. Neidhardt, N. Nguyen, F. Ricci, L. Rook, H. Werthner, and M. Zanker. Observing group decision making processes. In *Proceedings of the tenth ACM conference on Recommender systems, RecSys'16*, 2016.

[6] A. Delic, J. Neidhardt, and H. Werthner. Are sun lovers nervous? - research note at enter 2016 etourism conference. Bilbao, Spain, 2016.

[7] I. Fernández-Tobías, M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador. Alleviating the new user problem in collaborative filtering by exploiting personality information. *User Model. User-Adapt. Interact.*, 26(2-3):221–255, 2016.

[8] D. Forsyth. *Group Dynamics*. Wadsworth Cengage Learning, 6th edition, 2014.

[9] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM international conference on Supporting group work*, pages 97–106, FL, USA, 2010.

[10] H. Gibson and A. Yiannakis. Tourist roles: Needs and the lifecourse. *Annals of tourism research*, 29(2):358–383, 2002.

[11] L. R. Goldberg. An alternative "description of personality": the big-five factor structure. *Journal of personality and social psychology*, 59(6):1216, 1990.

[12] A. Jameson. More than the sum of its members: challenges for group recommender systems. In *Proceedings of the working conference on Advanced visual interfaces*, pages 48–54, 2004.

[13] J. Masthoff. Group recommender systems: aggregation, satisfaction and group attributes. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 743–776. Springer, 2015.

[14] K. McCarthy, L. McGinty, B. Smyth, and M. Salamo. The needs of the many: a case-based group recommender system. *Advances in Case-Based Reasoning*, pages 196–210, 2006.

[15] J. Neidhardt, R. Schuster, L. Seyfang, and H. Werthner. Eliciting the users' unknown preferences. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 309–312, 2645767, 2014. ACM.

[16] J. Neidhardt, L. Seyfang, R. Schuster, and H. Werthner. A picture-based approach to recommender systems. *Information Technology & Tourism*, 15(1):49–69, 2015.

[17] T. N. Nguyen and F. Ricci. Supporting group decision making with recommendations and explanations. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 24th Conference on User Modeling, Adaptation, and Personalization (UMAP 2016)*, Halifax, Canada, 2016.

[18] L. Quijano-Sanchez, J. A. Recio-Garcia, B. Diaz-Agudo, and G. Jimenez-Diaz. Social factors in group recommender systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):8, 2013.

[19] J. A. Recio-Garcia, G. Jimenez-Diaz, A. A. Sanchez-Ruiz, and B. Diaz-Agudo. Personality aware recommendations to groups. In *Proceedings of the 3rd ACM conference on Recommender systems*, pages 325–328, NY, USA, 2009.

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

[21] M. Stettinger, A. Felfernig, G. Leitner, S. Reiterer, and M. Jeran. Counteracting serial position effects in the choicla group decision support environment. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 148–157, GA, USA, 2015.

[22] R. S. Tindale and T. Kameda. Social sharedness as a unifying theme for information processing in groups. *Group Processes and Intergroup Relations*, 3(2):123–140, 2000.

[23] H. Werthner, A. Alzua-Sorzabal, L. Cantoni, A. Dickinger, U. Gretzel, D. Jannach, J. Neidhardt, B. Pröll, F. Ricci, M. Scaglione, B. Stangl, O. Stock, and M. Zanker. Future research issues in IT and tourism. *J. of IT & Tourism*, 15(1):1–15, 2015.

[24] H. Werthner and F. Ricci. E-commerce and tourism. *Communications of the ACM*, 47(12):101–105, 2004.

[25] A. Yiannakis and H. Gibson. Roles tourists play. *Annals of tourism Research*, 19(2):287–303, 1992.

Choice-based recommender systems

Paula Saavedra
CITIUS
University of Santiago de Compostela
Santiago de Compostela, Spain
paula.saavedra@usc.es

Pablo Barreiro
CITIUS
University of Santiago de Compostela
Santiago de Compostela, Spain
pablobv70@gmail.com

Roi Durán
CITIUS
University of Santiago de Compostela
Santiago de Compostela, Spain
roiduram@gmail.com

Rosa Crujeiras
School of Mathematics
University of Santiago de Compostela
Santiago de Compostela, Spain
rosa.crujeiras@usc.es

María Loureiro
School of Business
University of Santiago de Compostela
Santiago de Compostela, Spain
maria.loureiro@usc.es

Eduardo Sánchez Vila
CITIUS
University of Santiago de Compostela
Santiago de Compostela, Spain
eduardo.sanchez.vila@usc.es

ABSTRACT

Choice-based models are proposed to overcome some of the limitations found in traditional rating-based strategies. The new approach is grounded on decision-making paradigms, such as choice and utility theories. Specifically, random utility models were applied in a recommendation problem. Prediction accuracy was compared with state-of-art rating-based algorithms in a gastronomy dataset. The results show the superior performance of choice-based models, which may suggest that real choices could bring more predictive power than ratings.

CCS Concepts

•Information systems → Collaborative filtering; Social recommendation;

Keywords

Choice models; Random Utility Models; Logit probabilities; Tourism

1. INTRODUCTION

Recommender systems are personalization tools aimed at suggesting relevant items on the basis of available information on items as well as decision-makers [5]. Broadly speaking, recommenders can be classified in two different categories. Content-based recommenders generate a profile for each decision-maker by considering items experienced in the past. The profile typically represents the preferences of the decision-maker, i.e the taste of the decision-maker on each item's attributes [2]. These preferences can be used to predict the utility of any given item by comparing them with the values of item's attributes. Collaborative recommenders, on the other hand, take advantage of previous ratings provided by the available decision-makers to predict the utility of any given user-item pair [6]. This approach has been widely adopted as it removes the burden of knowing and managing item attributes as well as their corresponding values.

Many algorithms and models have been proposed under the collaborative paradigm. Among them, two families have gained major attraction: neighborhood algorithms and latent factor models. The neighborhood approach was the first to implement to collaborative concept and became the reference model in this research area [9, 4]. The method consists on representing vectors of ratings on either the decision-maker or item space. The distance between any pair of these vectors determine the similarity between either the decision-makers or the items that these vectors represent. Individuals with similar rating's vectors are considered to possess similar tastes or preferences, while items are considered to have similar attributes. The latent factor strategy, in turn, attempts to explain ratings by means of characterizing both users and items with a limited set of factors. Factors are considered unknown variables that can be inferred from the ratings declared by the users. The inference or learning problem can be solved with factorization techniques. The classical factorization method is called Singular Value Decomposition (SVD) and was applied successfully to identify and reduce the number of relevant factors [10]. However, the method requires complete knowledge of the rating matrix and fill-in methods to populate sparse rating matrix come at a cost of inaccurate factor learning. Recently, new factorization techniques have been successfully developed that are capable of learning the factors from sparse rating matrices [7]. Each rating is explained by means of two vectors whose dimensions correspond with the set of latent factors. The first vector represents the item in terms of its degree of possession of each factor, while the second vector represent the decision-maker on the basis of her preference on each factor. These item and decision-maker vectors constitute a pair of

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

matrices whose values have to be inferred. The learning problem is solved by means of minimizing the regularized error on the set of known ratings.

Despite the success of current recommender systems, the experience with state-or-art approaches reveal some important limitations. First, the degree of performance of a recommender algorithm depends on the specific issues of the problem at hand. Therefore, heuristic models and trial-and-error methodologies are often used to look for the best solution for any given situation. The problem may be approached in a more theoretical and consistent way if recommenders were considered as agents predicting the decisions taken by decision-makers. Under this scope, the first limitation could be stated as follows: *(L1) Current state-of-art approaches are mostly based on heuristic models rather than decision-making theories.* Second, some popular paradigms assume a direct relationship between preferences and ratings: (1) the neighborhood approach considers that decision-makers with similar ratings on a set of items will have similar preferences, and (2) factorization techniques assume that ratings can be the result of a product between item's latent factors and decision-maker preferences about that factors. In these paradigms unobserved preferences are usually inferred from observed ratings. The issue here comes from the fact that ratings could be mostly explained by variables different to preferences. The quality of the item, the user-item context, and in general any factor involved during the process of experiencing the item, they all could provide more explanatory power about ratings than preferences do. Therefore, the second limitation could be described as follows: *(L2) Preferences are usually derived from ratings without any supporting evidence about the relationship between these variables.*

This work proposes choice-based recommender systems to overcome these limitations. The concept is grounded on choice and utility theory, where real choices replace ratings as the key data to learn the decision-maker's preferences as well as to make recommendations. The proposed models are then evaluated in the tourism domain with a gastronomy dataset that includes both choices and ratings. In what follows, the choice-based models are presented, the methods are described, and the models evaluated and compared against state-of-art rating-based algorithms. The discussion will comment on the results and highlight the major contributions of the paper.

2. CHOICE MODELS

2.1 Recommendation as a choice problem

The recommendation problem can be described as an optimization problem which consists on (1) estimating the utility of each item $a \in A$, the available item set, for any given decision-maker c, and (2) choosing the item a' that maximizes $U(c, a)$, the decision-maker utility on any item a [1]:

$$a' = \arg\max_{a \in A} U(c, a) \tag{1}$$

It is worth noting that this problem is conceptually the same as the one faced by the Rational Choice Theory, which aims at explaining economic behaviour under choice situations [11]. The theory states that a decision-maker will maximize her utility after satisfying some budget constraints. More formally, the decision-maker will choose alternative a' from a choice set A according to the following rule:

$$CR(A, \succeq) = \{a' \in A \parallel a' \succeq a, \forall a \in A\} \tag{2}$$

where CR stands for "choice rule" and the $\succeq$ operator denotes the relationship "preferred to, or at least as preferred as". Basically, it means that the chosen alternative will be the one from which the decision-maker shows a higher preference. The preference operator needs to be quantified to allow a numerical comparison between the alternatives.

The utility theory comes to the rescue to solve this problem. One of the axioms of this theory states that it is possible to define a utility function such that:

$$a \succeq b \Longleftrightarrow U(a) \geq U(b). \tag{3}$$

And then, the choice rule in equation 2 can be represented in terms of the utility function and a numerical operator:

$$CR(A, \geq) = \{a' \in A \parallel U(a') \geq U(a), \forall a \in A\}. \tag{4}$$

It is now clear that the new choice rule is mathematically equivalent to the recommendation problem described in equation 1:

$$\begin{aligned} a' = \arg\max_{a \in A} U(c, a) \Longleftrightarrow \\ CR(A, \geq) = \{a' \in A \parallel U(a') \geq U(a), \forall a \in A\}. \end{aligned} \tag{5}$$

As the recommendation problem can be understood as a choice prediction problem, then the powerful models and techniques developed in this field can be naturally applied to generate recommendations.

2.2 Choice models with random utility

The choice rule models how decision-makers take their decisions. However, the problem of predicting such decisions is a different task. In real problems the researcher does not have access to all the factors and variables that decision-makers include to estimate utilities. For a concrete individual c_n, the researcher only knows some attributes of the alternatives, labeled x_j for all a_j alternatives with $j \in \{1, \cdots, J\}$, and some attributes of the decision-maker, labeled z_n. A function that relates these observed factors to the decision-maker's utility can be specified. This function is denoted by $V_{nj} = V(x_j, z_n)$ and it is often called representative utility. It usually depends on parameters that are unknown and, therefore, they must be estimated.

Since there are aspects of utility that the researcher does not or cannot observe, $V_{nj} \neq U_{nj}$. Therefore, the utility can be decomposed as:

$$U_{nj} = V_{nj} + \epsilon_{nj} \tag{6}$$

where ϵ_{nj} captures the unknown factors that modify the utility and are not included in V_{nj}. This decomposition is fully general, since ϵ_{nj} is defined as simply the difference between true utility U_{nj} and the part of utility that the researcher captures in V_{nj}. Given its definition, the characteristics of ϵ_{nj}, such as its distribution, depend critically on the researcher's specification of V_{nj}. The researcher does not know ϵ_{nj} for all j and therefore these terms are considered random variables that allow the researcher to make probabilistic statements about the decision-maker's choice. The models derived under this assumptions are called random utility models (RUM) [8].

Now, the choice rule of equation 4, which is deterministic under the decision-maker perpective, becomes probabilistic under the perspective of the researcher. Then the rule for a decision-maker c_n choosing alternative a_i is:

$$CR(A, \geq) = \{a_i \in A \,\|\, \mathbb{P}_i \geq \mathbb{P}_j, \forall a_j \in A\} \quad (7)$$

and the probability $\mathbb{P}_i$ is estimated as follows:

$$\mathbb{P}(U_{ni} > U_{nj} \text{ for all } j \neq i) = \mathbb{P}(\epsilon_{nj} - \epsilon_{ni} < V_{ni} - V_{nj} \text{ for all } j \neq i). \quad (8)$$

If the joint density of $\epsilon_n = (\epsilon_{n1}, ..., \epsilon_{nJ})$ is denoted by f, this cumulative probability can be rewritten as:

$$\mathbb{P}_{ni} = \int_\epsilon \mathbb{I}(\epsilon_{nj} - \epsilon_{ni} < V_{ni} - V_{nj} \text{ for all } j \neq i) f(\epsilon_n) d\epsilon_n \quad (9)$$

where $\mathbb{I}$ is the indicator function, equaling 1 when the term in parentheses is true and 0 otherwise. This is a multidimensional integral over the density of the unobserved portion of utility, $f(\epsilon_n)$. Different choice models are obtained from different specifications of this density, that is, from different assumptions about the distribution of the unobserved portion of utility. In addition, the choice of the density determines whether the integral takes a closed form or not [12].

2.3 Standard and mixed logit models

The simplest and most widely used choice model is the standard logit model [8]. It is derived under the assumption that the each unobserved portion of utility ϵ_{nj} is distributed independently, identically extreme value. In this case, f denotes the density for Gumbel distribution:

$$f(\epsilon_{nj}) = e^{-\epsilon_{nj}} e^{-e^{-\epsilon_{nj}}}. \quad (10)$$

Following [8], the logit choice probability that decision-maker c_n chooses alternative i is

$$\mathbb{P}_{ni} = \frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}}. \quad (11)$$

This model presents a clear interpretation. According to equation 11, if V_{ni} rises, reflecting a matching between the observed attributes of the alternative and the preferences of the decision-maker, with V_{nj} for all $j \neq i$ held constant, $\mathbb{P}_{ni}$ approaches one. And $\mathbb{P}_{ni}$ approaches zero when V_{ni} decreases, since the exponential in the numerator approaches zero as V_{ni} approaches $-\infty$.

The representative utility is usually specified to be linear in the set alternative's attributes: $V_{nj} = \beta_{nj} \cdot x_j$, where x_j is a vector containing, as before, the observed variables of the alternative a_j, and β_{nj} denotes the model coefficients vector which describes the preferences of decision-maker c_n on the attributes of the alternatives a_j. The preferences β_{nj} (model coefficients) are estimated by fitting equation 11 to a dataset of choices. Moreover, since the logit probabilities take a closed form, maximum likelihood procedures are applied for estimation. Concretely, the probability of person c_n choosing the alternative that he was actually observed to choose can be expressed as

$$\prod_i \mathbb{P}_{ni}^{y_{ni}},$$

where $y_{ni} = 1$ if the individual choses i and zero otherwise. Since $y_{ni} = 0$ for non-chosen alternatives and $\mathbb{P}_{ni}$ raised to the power of zero is 1, this term is simply the probability of the chosen alternative. Assuming that decision-maker's choices are independent, the probability of each individual choosing the alternative that she was observed actually to choose is

$$L(\beta) = \prod_n \prod_i \mathbb{P}_{ni}^{y_{ni}}$$

where β denotes the vector of all model parameters. Therefore, the log-likelihood function is

$$LL(\beta) = \sum_n \sum_i y_{ni} \log \mathbb{P}_{ni}$$

and the estimator is the value of β that maximizes this function. Importantly, it was proved that the log-likelihood function with these choice probabilities is globally concave in parameters β, which helps in the numerical maximization procedures, see [8] for more details.

A well-known issue of standard logit model deals with capturing the heterogeneity of population [12]. The importance that decision-makers place on each attribute of the possible choices varies, in general, over decision-makers. Although logit model is able to represent the taste variation related to observed characteristics of the decision-maker, it can not represent differences in tastes that can not be linked to observed characteristics. Therefore, if taste variation is at least partly random, a logit model with random parameters should be considered instead. Under this considerations, β is now a vector of random coefficients and these coefficients vary over decision-makers in the population with density g. In most applications that have actually been called mixed logit, g is specified to be continuous. For example, it can be specified to be normal, lognormal, uniform, triangular or, even, gamma. Therefore, this density is a function of parameters θ that represent, in the gaussian case, the mean and covariance of the random coefficient in the population. Then, the choice probabilities can be written as:

$$\mathbb{P}_{ni} = \int \left(\frac{e^{V_{ni}(\beta)}}{\sum_j e^{V_{nj}(\beta)}} \right) g(\beta|\theta) d\beta. \quad (12)$$

Since the previous integral has not a closed form, it must be evaluated numerically through simulation. Once the researcher specifies a distribution g for the coefficients, the parameters θ maximizing the simulated log-likelihood must be estimated. Then, R draws of the coefficients are taken from g and the logit probabilities are computed for every draw. The unconditional probability in equation 12, that is the expected value of the conditional probabilities, is estimated as the average of R probabilities determined previously.

3. METHODS

The performance of choice-based models is compared with a choice of relevant rating-based algorithms from a gastronomic dataset containing the choices of snacks made by a set of decision-makers and their corresponding tapa ratings. The dataset is described in Sections 3.1 and 3.2. Technical details on the two recommendation alternatives considered in this work are briefly presented in Sections 3.3 and 3.4. Finally, the error criteria used to compare them are introduced in Section 3.5.

3.1 Experiment

In the context of the RECTUR project, an experiment was carried out with real users in the context of Santiago(é)Tapas, a gastronomic context that takes place every year in Santiago de Compostela. In 2011 the fourth edition was held with a total of 56 participating restaurants proposing and elaborating up to three tapas that were sold at a price of 2 euro. The experiment was designed to gather relevant data while preserving the spirit of the contest. Participants were local users as well as Spanish and international tourists. A TapasPassport with the official information about the contest was made available to all participants. It contained: (i) the contest guidelines and other related information to the participants, (ii) restaurants location, (iii) the tapas offered on each restaurant, (iv) an official seal to demonstrate that a participant has visited the minimum number of restaurants required to obtain contestś gifts. Restaurant staff had to sign the TapasPassport to certify that its owners have visited the place.

After consuming a tapa, participants were asked to evaluate their experience. Users had to provide two ratings ranging from 0 to 5: (i) a rating of the tapa, and (ii) a rating of the overall experience (service, place atmosphere, etc.). In addition, they were informed about our research experiment and asked to extend their feedback providing information about the temporal and social context in which the experience took place.

3.2 RECTUR Dataset

The data gathered in the experiment was collected in the RECTUR dataset. It is assumed that the choice of a tapa depends on the user preferences about the levels of tapa attributes, which will in turn depend on the user attributes and context elements. The consumption of a tapa determines a choice from a choice set and will elicit a satisfaction response quantified as a user rating.

For each tapa, we gathered the following attributes:

- Choice sets. Different choice sets could be defined for each choice. We acquired information about the following sets:
  - Set of tapas in the same area of the city (outlying, new or old zone).
  - Set of tapas in the same restaurant.
- Tapa attributes. The gathered attributes are:
  - Type: Cheese, egg, fish, meat, vegetable, shellfish and other. The main ingredient defined the type of the tapa.
  - Character: Traditional or daring. Traditional tapas are those that follow popular well-known recipes, while daring tapas are creative and provide innovative recipes.
  - Restaurant. The restaurant that offers the tapa was also categorized in terms of its location (outlying, new or old area), atmosphere and style.
- Rating. The rating provided by each consumer.

3.3 Choice-based models

The standard logit model as well as the mixed logit model assuming Gaussian distribution on the coefficients, both described in Section 2.3, were chosen as basic representatives of the family of random utility choice-based models to be compared with rating-based algorithms. From attributes type and character of each tapa described in Section 3.2, eight binary variables associated to each alternative (or snack) were generated for fitting these two models. Next, the construction of the variables is briefly described through an example. The choice set associated to the old area contains, as possible choices, the set of tapas distributed in restaurants of this zone. For each one of these snacks, the dichotomous variables cheese, egg, fish, meat, vegetable, shellfish and traditional are generated. According to Figure 2, the main ingredient of t100 is meat. However, this tapa is not traditional. Therefore, only the variable meat will be equal to 1. The rest of variables associated to t100 will take the value zero.

Within the discrete choice framework, the set of alternatives known as the choice set must verify three properties. It has to be finite, exhaustive (the decision-maker always chooses one of the alternatives) and mutually exclusive (the choice of one alternative necessarily implies not choosing any of the other ones). Due to the last property, three different choice subsets were established in this work. They correspond to the three possible restaurant locations (old, new and outlying areas of the city). Therefore, standard and mixed logit models are estimated separately from these three choice subsets that contain only the tapas associated to each zone. This assumption could be less general. For instance, considering the set of tapas of a concrete restaurant would provide a new choice set and, as consequence, a new choice problem.

Estimations results for these six models are shown in Section 4.2. For the same area of the city, standard and mixed logit models present similar estimations for the coefficients. As consequence, only prediction accuracy of the standard logit model was compared with rating-based algorithms.

3.4 Baselines: Rating-based models

The proposed choice-based models were compared with two popular rating-based models: User-based collaborative filtering (UBCF) and matrix factorization (MF). User-based collaborative filtering assumes that individuals with similar preferences will rate items in a similar way. Then, missing ratings for a concrete user c_n could be predicted finding a neighborhood $N(n)$ of similar users and aggregating their ratings to calculate the corresponding prediction. The concept of similarity between users is used for defining this neighborhood given all users within a similarity threshold. In this work, the cosine similarity measure is taken into account and $|N(n)|$ was fixed equal to 25. For an item i and an individual c_n, the ratings predicted, $\hat{r}_{ni}$, can be written as

$$\hat{r}_{ni} = \frac{1}{|N(n)|} \sum_{j \in N(n)} r_{ji}$$

where $|\ |$ denotes the cardinal of $N(n)$.

Matrix factorization, on the other hand, characterizes both items and users by vectors of factors inferred from item rating patterns. For a given item i and a user c_n, the vector q_i measure the extent to which the item possesses those factors and the vector p_n, the extent of interest the user has in items that are high on the corresponding factors. The dot product $q_i^T p_n$ captures the user's interest in the item's characteristics. This approximates user c_n's rating of item i, r_{ni}, leading to the estimate

$$\hat{r}_{ni} = q_i^T p_n.$$

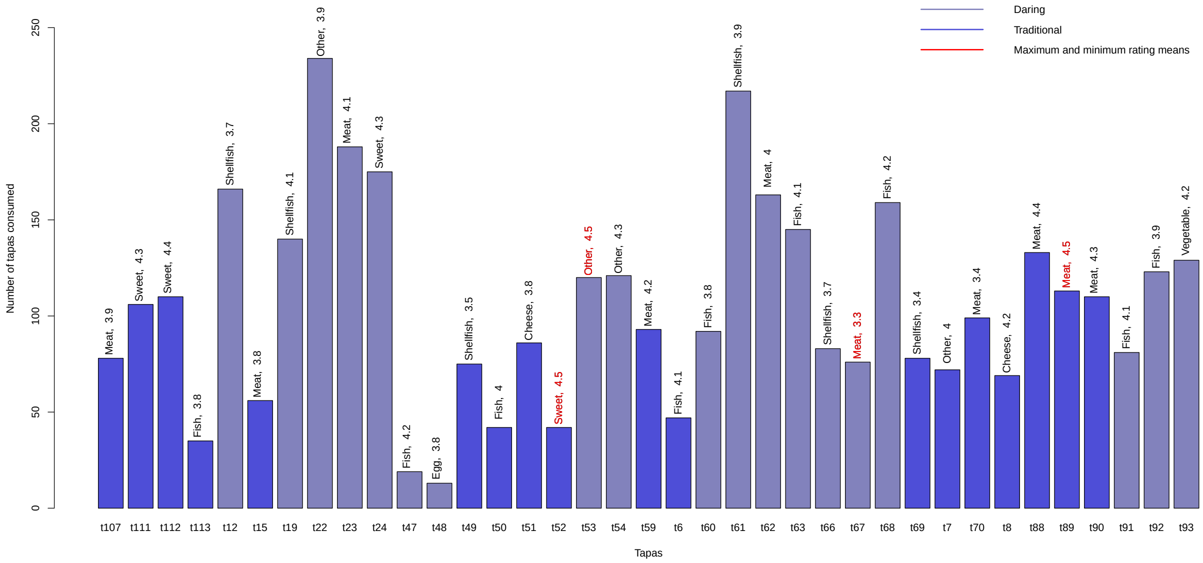


Figure 1: Bar plot for number of different tapas consumed, main ingredient and mean of users' ratings in the new zone of the city.

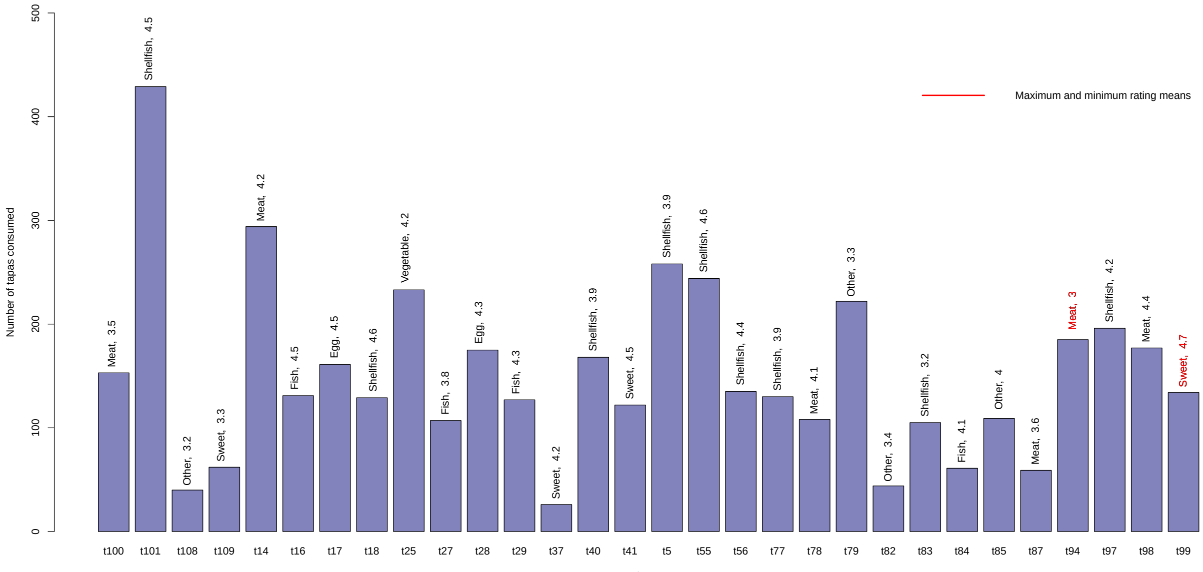


Figure 2: Bar plot for number of different daring tapas consumed, main ingredient and mean of users' ratings in the old zone of the city.

Therefore, the challenge is computing the mapping of each item and user to vectors q_i and p_n. Here, singular value decomposition will be applied factoring the user-item rating matrix that could be sparse. In order to learn the factor vectors (p_n and q_i), the regularized squared error on the set of known ratings is minimized:

$$\min_{q*,p*} \sum_{(u,i)\in K} (r_{ni} - q_i^T p_n)^2 + \lambda(\|q_i\|^2 + \|p_n\|^2)$$

where K is the set of the (c_n, i) pairs for which r_{ni} is known, $\| \; \|$ is the Euclidean norm and λ denotes a constant controlling the extent of regularization. In this work, $\lambda = 1.5$.

3.5 Evaluation

Classical ranking error metrics could not be applied mainly because of the lack of information about all the relevant tapas for the decision-maker on any choice situation. Therefore, two error metrics are proposed in order to compare the behaviour of choice-based and rating-based algorithms. The metrics are described considering that only the tapa

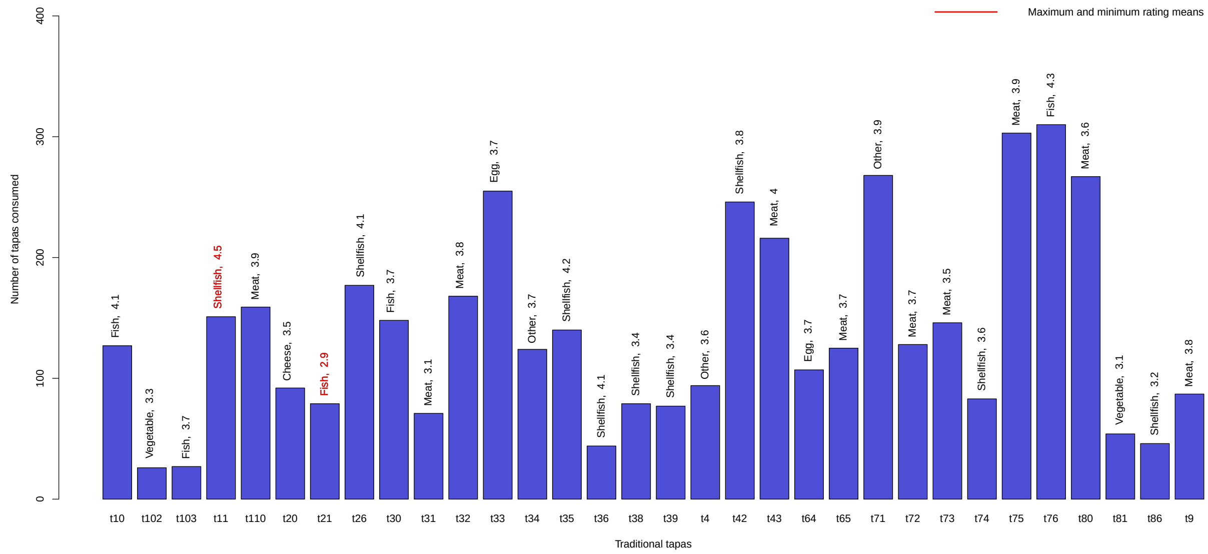


Figure 3: Bar plot for number of different traditional tapas consumed, main ingredient and mean of users' ratings in the old zone of the city.

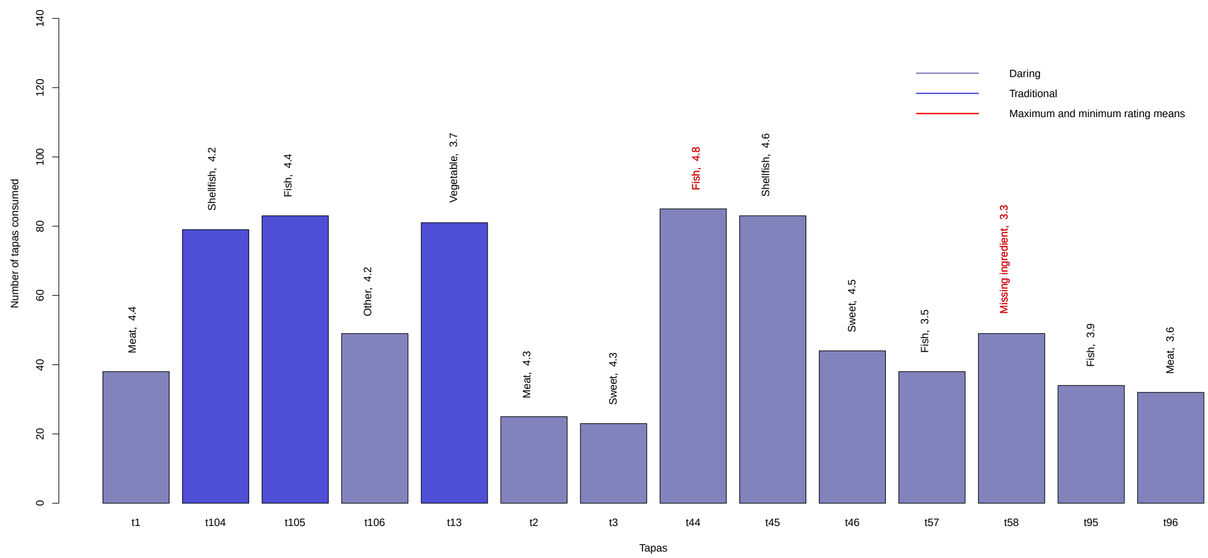


Figure 4: Bar plot for number of different tapas consumed, main ingredient and mean of users' ratings in the outlying zone of the city.

with the highest associated rating or probability is recommended/predicted (top 1). Error I is equal to one if the item predicted does not coincide with the true alternative chosen by the individual and zero otherwise. Therefore, given an individual c_n, the true choice i and the recommended item j is:

$$\text{error I } (c_n, i) = \begin{cases} 1 & : \text{ if } i \neq j \\ 0 & : \text{ otherwise.} \end{cases}$$

The second measure of error, error II, is equal to the position of the real choice in the ordered list of recommendation minus one. Therefore, if the item recommended is equal to the chosen one then the error is equal to zero. Let $(i_1, ..., i_k, ..., i_J)$ be the list of ordered items to be recommended, the error for the user c_n with true choice i can be written as:

$$\text{error II } (c_n, i) = k - 1 \text{ if } i_k \neq i.$$

For instance, if one decision-maker c_n chose the snack t1 among the snacks (t1, t104, t105, $\cdots$) and the prediction (ordered according to the highest ratings or probabilities) is equal to (t105, t104, t1, $\cdots$), then error I $(c_n, \text{t1}) = 1$. However, error II $(c_n, \text{t1}) = 2$.

Error I and error II can be generalized easily if a list of a concrete number of ordered items (in terms of probabilities or ratings) is recommended instead of recommending only one alternative. These two errors are equal to zero if, for an individual c_n, the true choice i belongs to the recommended list of items. Otherwise, error I will take the value one and error II, the position of the true choice i in the ordered list of non-recommended alternatives. In this work, a list of five alternatives will be considered (top 5).

4. RESULTS

4.1 Data description

RECTUR dataset presented in Section 3.2 deals with 5517 individuals, that make one or a sequential choices of one tapa among a set of 113 tapas distributed in Santiago de Compostela. Acording to comments in Section 3.3, three subsets of the original dataset will be considered distinguishing three different choice contexts or, equivalently, three zones of the city.

Next, the three scenarios will be briefly described.

The total number of tapas consumed in *new area of the city* is 3888. However, the number of different tapas associated to this zone is only 37; 18 of them present a traditional character and 19, a daring character. Furthermore, the number of users in this area is 2030. Then, although most of these individuals had only one snack, some of them took several ones. Figure 1 shows the total number of tapas that users consumed for the 37 possible choices. According to the results, t22 and t61 were the most common choices. However, t47 and t48 were rarely selected. According to the information in Figure 1, only one tapa is made of eggs and vegetables; two tapas has cheese as main ingredient; four tapas are made of a sweet component or other; shellfish is the ingredient of six snacks; meat and fish are the most common components with ten and nine tapas, respectively. Tapa ratings that users gave to one consumed tapa are available too. The values of these ratings are 0, 1, 2, 3, 4 and 5. High values for ratings are associated to a high customer satisfaction. Means of tapa ratings for the 37 tapas in the new area are shown in Figure 1. The lowest means of tapa ratings are associated to t67, t70, t69 and t49. However, all of these means are greater than 3. So, the level of satisfaction tends to be high.

As for the *old zone of the city*, a total of 8948 tapas were consumed. As before, the number of different snacks associated to this zone is only 62; 32 of them present a traditional character and 30, a daring character. Furthermore, the number of users in this area is 3953. As before, although most of these individuals had only one snack, some of them took several ones. Figures 2 and 3 show the total number of daring and traditional tapas that users consumed for the 62 possible choices, respectively. According to the results, t101 was the most common choice. However, t37, t103 and t102 were rarely selected. As regards means of snack ratings, the lowest ones correspond to t21 and t94. The highest ones, to t11 and t99.

The number of snacks consumed in the *outlying area of the city* is 743. Again, the number of different snacks associated to this zone is smaller. Concretely, it is equal to 14; 3 of them present a traditional character and 11, a daring character. Furthermore, the number of users in this area is 436. Figure 4 shows the total number of daring and traditional tapas that users consumed for the 14 choices. According to the results, t44, t45, t104 and t105 were the most chosen snacks. However, t2 and t3 were rarely selected. The snacks t58 and t44 correspond to the tapas with lowest and highest means of ratings, respectively. The main ingredient of t58 is a missing value. In addition, cheese and egg are not the main component for any snack.

4.2 Choice models fitting

The standard and mixed logit models have been fitted from the three choice sets described in Section 4.1. Due to the price is the same for every snack, the determinants of these choices, x_j, are eight dichotomous alternative specific variables. Seven of them indicate the main component of each tapa: Cheese, egg, fish, meat, shellfish, sweet and vegetable. The eighth variable takes value equal to one when the snack has a traditional character. In addition, for mixed logit model, Gaussian distribution was assumed on the coefficients and $R = 100$ was fixed.

	New zone	Old zone	Outlying zone
Cheese	-0.07	**-0.25**	
Egg	**-2.48**	**0.31**	
Fish	**-0.46**	-0.02	0.14
Meat	0.06	**0.28**	**-0.44**
Shellfish	-0.03	**0.21**	**0.38**
Sweet	0.07	**-0.46**	**-0.38**
Vegetable	**-0.18**	**-0.17**	0.26
Traditional	**-0.62**	**-0.15**	**0.24**
Log-Likelihood:	-13772	-36757	-1913.8

Table 1: Estimation by maximum likelihood of the standard logit model coefficients for different areas of the city. Significant coefficients are in black.

	New zone	Old zone	Outlying zone
Cheese	-0.07	-0.24	
Egg	**-2.48**	**0.31**	
Fish	**-0.46**	-0.01	0.13
Meat	-0.07	**0.27**	**-0.67**
Shellfish	-0.03	**0.21**	**0.37**
Sweet	-0.003	**-0.46**	**-0.38**
Vegetable	**-0.18**	**-0.17**	0.26
Traditional	**-0.93**	**-0.09**	-0.01
Log-Likelihood:	-13631	-36680	-1897.9

Table 2: Estimation of the means for mixed logit model coefficients assuming normal distribution for different areas of the city. Significant coefficients are in black.

The coefficients obtained are shown for each area of the city in Tables 1 and 2, respectively, and most of them are significant in the three areas of the city. For the mixed logit

	Choice model		UBCF		MF	
	Top 1					
Error	CV_1	CV_2	CV_1	CV_2	CV_1	CV_2
I	0.895	0.876	1	1	1	1
II	5.057	4.741	8.885	9.006	8.868	8.824
	Top 5					
I	0.408	0.409	1	1	1	1
II	1.841	1.859	6.262	6.295	6.128	6.115

Table 3: Cross validation predictions errors for standard logit choice model, user-based collaborative filtering and matrix factorization algorithms in the outlying area of the city. Random and leave-one-out cross validation are denoted by CV_1 and CV_2, respectively. In this zone, the number of different tapas to be recommended is 14.

	Choice model		UBCF		MF	
	Top 1					
Error	CV_1	CV_2	CV_1	CV_2	CV_1	CV_2
I	0.955	0.954	1	1	1	1
II	14.552	14.060	25.438	25.475	25.511	25.499
	Top 5					
I	0.795	0.789	1	1	1	1
II	10.606	10.481	22.606	22.640	22.658	22.506

Table 4: Cross validation predictions errors for standard logit choice model, user-based collaborative filtering and matrix factorization algorithms in the new area of the city. Random and leave-one-out cross validation are denoted by CV_1 and CV_2, respectively. In this zone, the number of different tapas to be recommended is 37.

	Choice model		UBCF		MF	
	Top 1					
Error	CV_1	CV_2	CV_1	CV_2	CV_1	CV_2
I	0.982	0.987	1	1	1	1
II	27.005	26.921	43.908	43.862	43.843	43.787
	Top 5					
I	0.905	0.904	1	1	1	1
II	23.141	23.097	41.013	40.964	40.945	40.970

Table 5: Cross validation predictions errors for standard logit choice model, user-based collaborative filtering and matrix factorization algorithms in the old area of the city. Random and leave-one-out cross validation are denoted by CV_1 and CV_2, respectively. In this zone, the number of different tapas to be recommended is 62.

model (Table 2), only the mean estimations of Gaussian distributions are shown. As for the utility, positive coefficients, see egg and meat in Table 1 for the old zone, increase its value. However, negative coefficients, see egg and traditional in Table 1 for the new area, reduce it.

4.3 Choice-based vs rating-based predictions

The behaviour of choice-based and rating-based models for recommending tapas in the three areas of the city was analyzed using random sub-sampling and leave-one-out cross validation from RECTUR dataset.

For random sub-sampling validation, 100 iterations were considered using the 25% of randomly selected individuals as test data for predictions. Therefore, in each iteration and once the 25% of decision-makers was randomly selected, the rest of individuals is used as trainning data for rating-based algorithms or for fitting the choice model. Then, for each decision-maker in the test data and for each recommendation method, prediction error measures introduced in Section 3.5 can be determined. The procedure for leave-one-out cross validation is similar. In this case, the number of iterations is equal to the number of users and, in each iteration, the test data contains an only decision-maker.

Tables 3, 4, and 5 contain the empirical means of errors decribed previously for the new, old and outlying areas of the city, respectively. According to results shown in Section 4.2, standard and mixed logit models provide similar estimations for model coefficients. Therefore, only the first choice-based model, the standard logit one, were taken into account to be compared with the rating-based algorithms.

The results show that choice-based models offer a better performance (lower prediction errors) compared with rating-based schemes (UBCF and MF). See, in particular, error II for the top 5 scheme taking into account the different number of tapas recommended in each area of the city. Furthermore, the accuracy of predictions is reduced as long as the choice set increases from the outlying to the old area, which indicates the importance of the choice set and the choice situation.

5. DISCUSSION

The main point of this work is that the recommendation problem can be considered as a choice prediction problem. This is the main difference of our proposal compared with current paradigms in recommender systems that focus on rating prediction. The key aspects of our choice-based models are: (1) preferences are learnt from choices, (2) the choice set of each choice situation is included as a relevant variable to both explain and predict future choices, and (3) unobserved factors affecting the decision-making process are captured through random variables. On the basis of these elements the models presented in this paper differ from both collaborative methods, as they infer preferences from ratings, and content-based techniques, as they do not handle the choice set of the items experienced in the past. Recent content-based approaches share the same idea about the utility of user choices to derive preferences but are limited to pairwise rather than full choice set comparisons [3].

With regard to the limitations stated in the introduction, choice models face issue *L1* by building random utility models from solid decision-making theories, and solve issue *L2* by using choices, rather than ratings, to estimate preferences. The drawback of gathering information about the domain (attributes and values) is compensated in two ways: (1) by using more accurate data, choices rather than ratings, and (2) by removing the burden of interrogating decision-makers about their post-experience satisfaction. In summary, choice modelling seems to be a promising paradigm in the field of recommender systems.

Acknowledgments

This research was sponsored by EMALCSA/Coruña Smart City under grant CSC-14-13, the Ministry of Science and Innovation of Spain under grant TIN2014-56633-C3-1-R, and the Ministry of Economy and Competitiveness of Spain under grant MTM2013-41383P.

6. REFERENCES

[1] T. A. Adomavicius G. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005.

[2] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[3] L. Blédaité and F. Ricci. Pairwise preferences elicitation and exploitation for conversational collaborative filtering. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 231–236. ACM, 2015.

[4] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[5] G. M. Burke R., Felfernig A. Recommender systems: An overview. *AI Magazine*, 32(3):13–18, 2011.

[6] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[7] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[8] D. McFadden et al. Conditional logit analysis of qualitative choice behavior. 1973.

[9] V. H. Resnick P. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[10] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

[11] A. Sen. Rational behaviour. In *Utility and probability*, pages 198–216. Springer, 1990.

[12] K. E. Train. *Discrete choice methods with simulation.* Cambridge university press, 2009.

Anything Fun Going On? A Simple Wizard to Avoid the Cold-Start Problem for Event Recommenders

Stacey Donohue
relEVENTcity
stacey@releventcity.com

Nevena Dragovic & Maria Soledad Pera
Dept. of Computer Science
Boise State University,Boise, ID, USA
{ndragovic,solepera}@boisestate.edu

ABSTRACT

In this demo, we showcase a set up wizard designed to bypass the cold start problem that often affects recommendation systems in the event domain. We have developed a mobile application for tourists, RelEVENT, which allows them to quickly and non-intrusively set up preferences and/or interests related to events. This will directly affect the degree to which they can receive personalized recommendations on-the-fly and become aware of events happening around town that might be appealing to them.

CCS Concepts

•Information systems → Decision support systems; Recommender systems; *Personalization;*

Keywords

Recommendation system; cold start; events; wizard; mobile application

1. INTRODUCTION

Recommendation systems can help users in locating items (e.g., products and services) of interest more quickly by filtering and ranking them based on some criteria, i.e., location, popularity, or preference, to name a few [1]. No matter if it is related to shopping web-sites (Amazon, e-bay, cheapoair, etc.), news related web-sites (Yahoo, CNN, etc.), hotel search or restaurant search, recommendation systems have a huge influence on businesses success and users' satisfaction. Thanks to those systems, companies and products are able to get advertisement by being offered to potential buyers. At the same time, recommenders enhance users' experience by assisting them in finding information pertaining to their preferences.

Recommenders focusing on common products or services, such as books, movies, or restaurants, have been well-studied and developed. However, research efforts related to recommendations within the tourism domain are less prolific and must address novel challenges pertaining specifically to this domain [3]. In fact, most existing works in this domain focus on suggesting specific places or events. Places are often associated with well-known geographical locations, i.e., Points-of-interest (PoI) [4], such as the Eiffel Tower or New York Yankees Stadium. Events, on the other hand, are usually short-lived and varied in nature. Within the tourism domain, events pose a special challenge for recommendation strategies given the lack of uniform event metadata and historical information in the form of personal ratings. Events are varied in nature, ranging from concerts and sports games to small gatherings or dinner parties and can occur in diverse locations that can often change and do not necessarily correspond to a PoI.

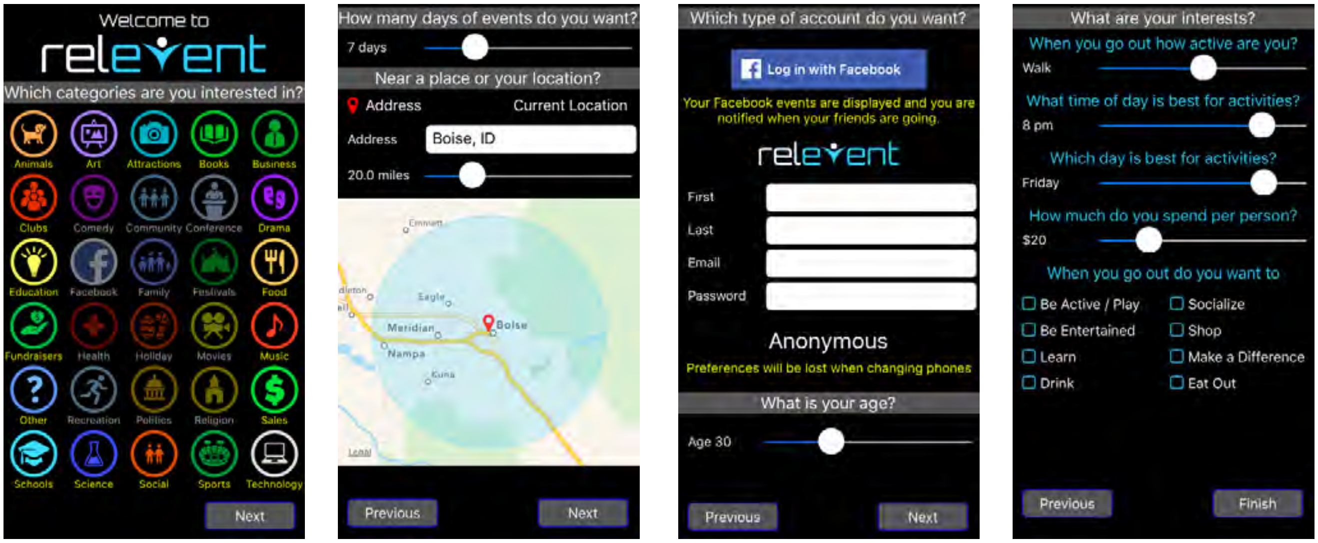
Regardless of the domain, cold start is one of the most "popular" challenges that hinders all recommendation systems. Cold start occur when the system is not able to create recommendations due to unavailable historical data for new users or items. This can be the reason why recommenders cannot be more successful in creating personalized suggestions and linking items to users. The cold start challenge is even harder to solve in the case of suggesting events. This is due to the fact that events have short time span and cannot be recommended after they end [2]. While this complicates the issue from an event perspective, we can address this problem by focusing on the users instead.

In this demo, we present wizard used by RelEVENT, the mobile recommendation application we developed, for bypassing the cold start problem in suggesting events at specific cities that people may find useful or interesting during their visit. The goal of this wizard is to collect enough data a-priory to provide personalized suggestions without imposing too much burden on the users. While the idea of a wizard is not unique to RelEVENT, to the best of our knowledge, our strategy is the first one that offers a balance of initial information to personalize suggestions and differs from strategies in the tourism domains, such as the one presented in [Bor15], which focus on type of traveler group, age, date, and motivation. We are aware that some users may prefer to bypass such a wizard, in which case the default options will still aid RelEVENT in providing suggestions tailored to proximity and popularity, i.e., provided suggestions will relate to the most popular events at that time in a given city.

2. OVERVIEW OF THE SYSTEM

RelEVENT includes the wizard made a specific set of questions that helps RelEVENT in filtering events for new users and avoiding the cold start problem and offer on-the-fly suggestions.

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



(a) Categories (b) Location (c) Demographic (d) Context

Figure 1: Snapshots of RelEVENT setup wizard

As shown in Figure 1(a), initially, our application will ask a user to select a number of well-known categories of interest. In addition, we included Facebook as a special category to allow users to include in their list of possible events to be recommended events that are publically available on Facebook. In doing so, our application can not only recommend the more "typical" events happening around a specific location, from conferences to movies to sales, but can also focus on more spontaneous and unique events, such a technical group meeting, e.g., ACM-W meeting at a university or âĂę

Tourists can be visiting a location for a short period of time for an extended vacation. With that in mind (as illustrated in Figure 1(b)), by default, a user will receive recommendations occurring within seven days. However, if desired, they will have the opportunity to decrease or increase the range of dates from which recommendations will be generated. A key aspect of recommendations related to tourism is distance. Users may favour events within close proximity or may be willing and able to move farther around town. Our application uses by default a 20 miles radius to limit the locations where events to be suggested occur. This radius can be adjusted by each user based on their own preferences and needs.

Age (shown in Figure 1(c)) is another dimension considered by RelEVENT. While not novel, it is one of the simplest questions that will help the recommender engine eliminate from their set of candidate events to recommend those that do not target the demographic of the user.More importantly, it will help eliminate from the list of possible recommendations those pertaining to events that occur where minors cannot attend.

As shown in Figure 1(d), the most interactive set of questions appear at the end of the wizard. RelEVENT is interested in finding out, if possible, the context or type of activities a visitor has in mind. In doing so, the recommender the recommender engine will be able to further narrow down the options available for each user and thus further personalize the provided recommendations. While Level of activity and Overall intention of events will lead to suggestions that match the physical abilities and expectations of each user, the time, date, and budget will ensure that suggested activities are appealing to each users.

3. CONCLUSION

In this demo we described the solution we implemented to deal with the cold start problem affecting event recommendation systems. To provide the most relevant suggestions to each user, we created a short wizard that will allow new users to off RelEVENT enough information about their interests and preferences to instantaneously receive appealing suggestions. Based on initial testing and feedback collected from users, we are encouraged with the performance and usability of wizard.

4. REFERENCES

[1] F. Gedikli, D. Jannach, and M. Ge. How should i explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382, 2014.

[2] H. Khrouf and R. Troncy. Hybrid event recommendation using linked data and user diversity. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 185–192. ACM, 2013.

[3] A. Moreno, L. Sebastiá, and P. Vansteenwegen. Tours'15: Workshop on tourism recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 355–356. ACM, 2015.

[4] Y.-T. Zheng, Z.-J. Zha, and T.-S. Chua. Mining travel patterns from geotagged photos. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):56, 2012.

“One Size Doesn’t Fit All": Helping Users Find Events from Multiple Perspectives

Sean MacLachlan & Stacey Donohue
relEVENTcity
{sean,stacey}@releventcity.com

Nevena Dragovic & Maria Soledad Pera
Dept. of Computer Science
Boise State University,Boise, ID, USA
{ndragovic,solepera}@boisestate.edu

ABSTRACT

In this demo, we showcase a novel mobile application that offers various ways to present recommendations to users. While the majority of the existing applications in the tourism domain either focus on event recommendation or event browsing, our mobile application acknowledges the fact that users have different interests at different times and for different occasions. Consequently, while suggested events are filtered and ranked by proximity and date ranges to ensure they suit users' needs, each user is allowed to choose how to access these suggestions in one of four ways: search, categorized browsing, following, and traditional recommendations.

CCS Concepts

•Information systems → Decision support systems; Recommender systems; *Personalization;*

Keywords

Event recommendations; user perspectives; mobile application

1. INTRODUCTION

As defined in [1], recommendation strategies in the tourism domain can help users, i.e., visitors, find unique and interesting information about a particular travel destination that match their preferences and their current context.

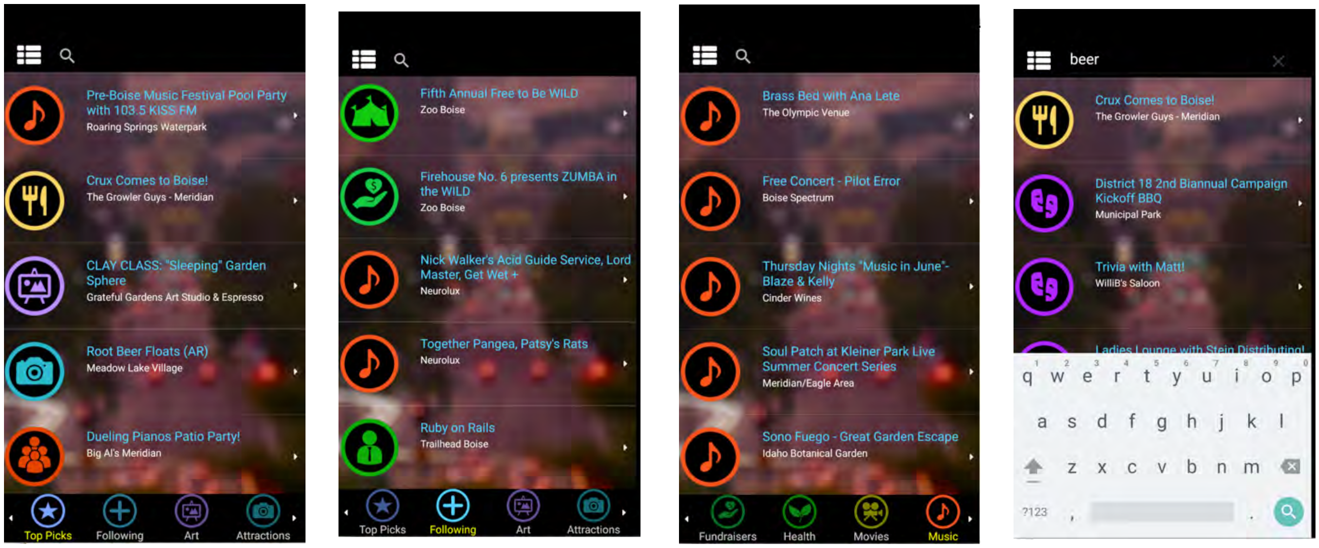
A recent survey on recommenders in the tourist domain [1] highlights the fact that the majority of the mobile applications focused on helping visitors create routes or tour plans, which often involved suggestions primarily focused on Points of Interest [3] and locations. Unlike points of interests, which are locations or places that people tend to find interesting, such as a museum, theater, or historical site, events occurring around town often have a short time-span, information (reviews and ratings) about them is usually limited, and they rarely reoccur. For example, the Eiffel tower is a well-known and popular tourist location with few restrictions besides the visiting hours and distance (which impacts transportation). Events, however, are more complicated. A concert or dinner event will also have time and location constraints, but the limited duration of the event combined with the lack of any historical data such as reviews or rating (because events often do not repeat) makes event recommendations much more challenging. Even after addressing the challenges inherent to event recommendation strategies, the application still needs to address the fact that different users have different preferences in terms of how they receive recommendations.

Based on our user analysis, we noticed that users not only differ in terms of the type of recommendations they favor, they can also prefer different types of recommendations depending on the circumstances (time of day, day of the week, weather, current mode, etc.). In the end, providing lists of “things to do”, even if they all appeal and are tailored to individual users is not enough. Context plays a key role, and it is the duty of the recommender to both narrow down choices and provide enough flexibility to cater to users' diverse needs. With this in mind, we present RelEVENT, a mobile application that offers different recommendation styles, thus allowing users to choose the one that best fits their current, but likely to change, information retrieval preference. Our analysis revealed four different groups of users: (i) the ones that consider only suggestions provided by the app and do not feel the need for further exploring, (ii) the ones that are interested in events their friends are going, or their favorite band is organizing, (iii) the ones that are only interested in one type of event at time (Friday for dinners, Saturday for sports), and (iv) the ones that are already aware of events they would like to visit but they need more information about them.

2. OVERVIEW

In this section we discuss each RelEVENT's strategies to provide recommendations targeted to users' needs. **Traditional Recommendations.** The main contribution of RelEVENT is providing personalized event suggestions for each individual user. The algorithm running in the background matches users' preferences, demographical information and historical data (such as, ratings, reviews, likes), with available metadata of each candidate event happening in a given city (Figure 1a). Based on our recommendation strategy, each user is provided with top-N suggestions from which to choose. Users who like receiving diverse (sometimes even unexpected) suggestions, prefer this type of recommendation. An ancillary benefit to this approach is that it doesn't require any effort from the user in order to receive credible

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



(a) Traditional Recommendation (b) Follow (c) Category (d) Search

Figure 1: Screen captures of our mobile application, which illustrate the different manners in which users can take advantage of RelEVENT to identify suitable events

recommendations. Therefore, a significant group of users prefers this type of recommendation delivery method.

Follow. Based on our empirical study conducted to detect preferred styles of recommendations, we noticed that another group of users are interested in knowing about the events their friends like (e.g., movie that is currently playing in the nearby theater), are interested in (e.g., concert that will happen in 3 months) or already attended (e.g., museum exhibit). Since these users prefer seeing this type of recommendation, RelEVENT (as shown at Figure 1b) provides suggestions based on the people they follow. In addition to following a person, we added a feature that allows users to follow a specific event. For example, the Cannes movie festival occurs every year and has a wide variety of movie screenings. By following the event, users can keep up to date about changes in the festival lineup or smaller events happening within the bigger event (e.g., of director roundtables associated with the film screenings).

Category. Based on our study, we concluded that users often prefer one category of events over another at a specific point of time (e.g., time of day, day of week, or month). During the football season, we noticed some of the users are more interested in the sports category on Sundays, while on Friday night users preferred events related to movies and dinners. In order to provide suitable recommendations related to specific category, RelEVENT provides users the ability to search and filter suggestions based on the current category of interest as shown in Figure 1c. The suggestions in each category will still be ranked based on the specific interest of a user (e.g., football fans will find the Seahawks game ranked first in the sport sections).

Search. The last group contains users who already know what type of events they would like to attend. RelEVENT enables them to find information about different types of events by doing a basic keyword search. In doing so, users can submit to the app specific constraints and still retrieve events that are relevant to their specifications, yet they are sorted, i.e., recommended, based on what RelEVENT knows this user favors (in terms of location, budget, etc). As shown in the Figure 1d, if a user is interested in going to an event related to "beer" he can type that keyword and RelEVENT will locate events that refer to "beer" in their archived metadata, filter and personalize the identified events, and provide suitable and relevant suggestions to the user.

3. CONCLUSIONS

We have introduced a mobile application that allows users to discover interesting events from multiple perspectives. While, as stated in [2], context-aware venue suggestion is still a challenge for the RecSys and Information retrieval communities, by offering exploratory and traditional avenues for recommendations to ease users' decision making process and therefore represents a step forward. Based on data collected using the mobile application presented in this demo, we will conduct the necessary empirical analysis to validate and quantify the degree to which offering multiple perspectives can increase user satisfaction in the recommendation process and allow them to take better advantage of the events a new destination offers visitors.

4. REFERENCES

[1] J. Borras, A. Moreno, and A. Valls. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16):7370–7389, 2014.

[2] R. Deveaud, M.-D. Albakour, J. Manotumruksa, C. Macdonald, I. Ounis, et al. Smartvenues: Recommending popular and personalised venues in a city. In *ACM CIKM*, pages 2078–2080, 2014.

[3] Y.-T. Zheng, Z.-J. Zha, and T.-S. Chua. Mining travel patterns from geotagged photos. *ACM TIST*, 3(3):56, 2012.

Position Paper: Combining Mobility Services by Customer-Induced Orchestration

Jan Fabian Ehmke
Freie Universität Berlin
Garystr. 21
14195 Berlin, Germany
+49-30-838-58731
janfabian.ehmke@fu-berlin.de

Dirk Christian Mattfeld
Technische Universität Braunschweig
Mühlenpfordtstr. 23
38106 Braunschweig, Germany
+49-531-391-3210
d.mattfeld@tu-braunschweig.de

Linda Albrecht
Freie Universität Berlin
Garystr. 21
14195 Berlin, Germany
+49-30-838-58731
linda.albrecht@fu-berlin.de

ABSTRACT
This position paper discusses the customer-oriented combination of mobility services offered by multimodal mobility platforms. We present a process-oriented approach on the selection and provision of complex mobility services and give an overview of state-of-the-art mobility platforms in German-speaking areas. We exemplify the limitations of current mobility platforms with regard to customer-orientation and claim that these platforms do not consider travelers' preferences to a sufficient extent. Based on these results, we motivate the need for customer-induced orchestration platforms that support customers in combining mobility services with services of other domains. Contrary to operator-induced combination of services, customer-induced orchestration would allow customers the autonomous selection of component services and support their orchestration to bundles of mobility and complementary services.

CCS Concepts
• Applied Computing→Transportation • Applied Computing →Reference Models.

Keywords
Service selection; service platforms; reference models; customer context; customer-induced orchestration.

1. INTRODUCTION
Digitization and interconnectedness play an important role in the domain of mobility and transportation services. In recent years, traditional mobility services such as public transportation have been amended by innovative mobility services such as car and ride sharing. These innovative shared mobility services are characterized by their flexible spatio-temporal availability [1]. Flexibility is enabled through automated business processes that link travelers and service operators in a highly-efficient, automated manner. However, while the access to these innovative services is made as easy as possible through smartphone apps, the use of a single car sharing service, for example, is usually not sufficient to fulfill the demand of a traveler. For planning a trip from door to door, several *component* mobility services must be selected and combined to a *complex* mobility service, ideally considering complex preferences of the traveler [2].

Multimodal mobility platforms promise to integrate traditional, timetable-bound public transportation with innovative mobility services such as shared mobility services. In recent years, the number of multimodal mobility smartphone apps and online platforms has increased significantly. However, available platforms differ heavily in functionality and customer orientation. In this position paper, we discuss current functionality and limitations of mobility platforms based on an overview of existing platforms for German-speaking areas (see Sect. 2). We use the discovered limitations to motivate the need for a new paradigm, namely *customer-induced* orchestration of complex mobility and complementary services. The corresponding framework is discussed in Sect. 3 and extended to customer-induced orchestration of services beyond mobility services. The position paper is concluded in Sect. 4.

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

2. MULTIMODAL MOBILITY PLATFORMS
Multimodal mobility platforms promise to support the selection and bundling of mobility services to complex services bundles. To investigate the level of customer orientation that is already provided by current multimodal mobility platforms, we have analyzed their functionality and customer orientation with regard to the support of the mobility service process. Fig. 1 shows the mobility service process, which was derived by [4] from a generic service process scheme proposed by [3]. The mobility service process distinguishes five phases, namely *search for information*, *consulting*, *booking*, *realization* and *payment*. In an ideal setting, multimodal mobility platforms would facilitate the configuration and execution of complex mobility services for all phases of the mobility service process.

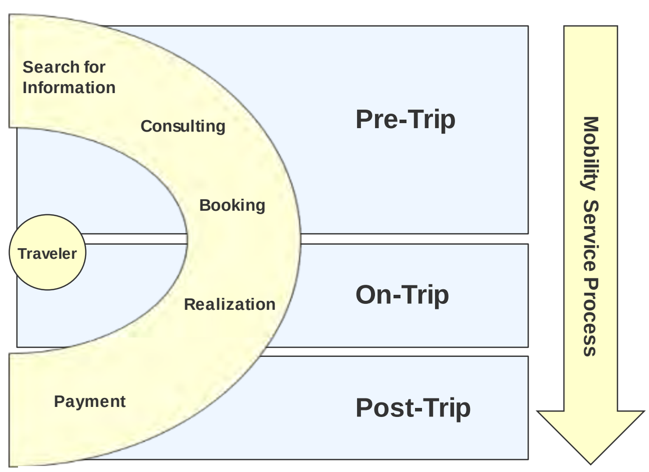


Figure 1. Mobility Service Process.

The *search for information* phase comprises all activities that help the traveler in discovering general information of available mobility services. The *consulting* phase discusses potential alternatives that could meet the traveler's needs, i.e., that consider the traveler's preferences. In the third phase, the traveler selects and *books* a particular mobility service, which is then *realized* by a mobility service provider. Finally, the traveler *pays* the stipulated fee for the utilized service. From a mobility research perspective, these phases can be understood as supporting the "pre-trip", the "on-trip" and the "post-trip" part of a journey with appropriate information.

Table 1. Selected Multimodal Mobility Platforms.

Provider	Type	URL	Local	Long-dist
Allryder	S	www.allryder.de	x	
FromAtoB	S	www.fromatob.de		x
GoEuro	S	www.goeuro.de		x
Mobility Map	P	www.mymobilitymap.de	x	
Moovel	E	www.moovel.com	x	
Qixxit	E	www.qixxit.de	x	x
Rome2rio	S	www.rome2rio.com	x	x
Route RANK	S	www.routerank.com		x
Waymate	S	www.waymate.de		x

Based on the above mobility service process and dimensions defined by the well-known architecture of integrated information systems (ARIS) [5], we have compared existing multimodal mobility platforms available in German-speaking areas. To analyze the platforms, we developed a criteria catalog including the five dimensions *organization, functionality, quality, data* and *technology* with a total of 22 criteria. We have limited our comparison to mobility platforms that are able to combine at least three different component services to a complex mobility service. The considered platforms as well as their main focus are summarized in Table 1. Five platforms are owned by innovative startups (Type = "S") and two by established mobility service providers (Type = "E"; *Moovel* belongs to the Daimler AG and *Qixxit* belongs to Deutsche Bahn AG, the main German train operator). One further platform is operated by a private person (Type = "P"). The columns "Local" and "Long-dist" identify the platform's focus, i.e., whether they claim to be the distinguished platform for information on and booking of local or long-distance mobility services.

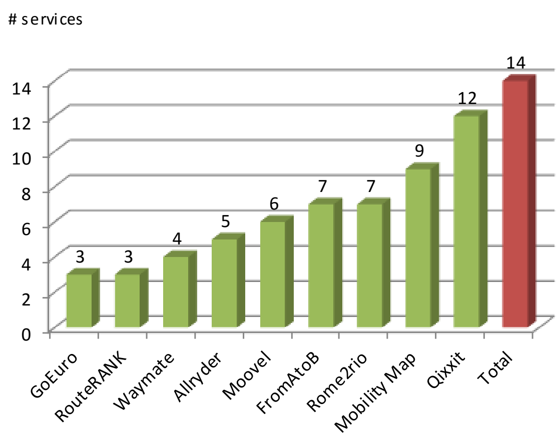


Figure 2. Component Services per Platform.

The number of considered component services per platform is shown in Fig. 2. The considered services are conducted by train, planes, taxis, rides, long-distance buses, private cars, local public transit, bike sharing, car sharing (free-floating/station based), walking, rental cars, private bike and ferry. The absolute number of considered component services varies significantly; platforms with a focus on local transport usually offer a larger number of component services than long-distance platforms. *Qixxit* provides the widest selection of component mobility services so far.

To investigate the functionality of the platforms, we developed the following three test instances reflecting a request for local travel, regional travel and long-distance travel:

- Local: Berlin/Breitscheidplatz to Berlin/Pariser Platz
- Long-Distance: Berlin/Main Station to Cologne/Main Station
- Regional: Karlsruhe/Main Station to Freiburg/Main Station.

To determine the appropriate complex mobility services that fulfill the above requests, each platform needs to combine component services according to travelers' preferences based on data such as expected time, location and price of a service. This is especially challenging for the long-distance request, where component services from different areas and of different modes need to be selected. Typically, at least one long-distance trip (e.g., per train or plane) and two local or regional trips (first/last mile to/from the long-distance trip) are to be combined. As a result, only six out of nine platforms are able to combine component services that operate on different modes (intermodal solutions), while the others are not able to augment long-distance with last-mile services. Travelers using these platforms end up in manually assembling their complex services by combining their preferred last-mile service with the help of other platforms or smartphone apps.

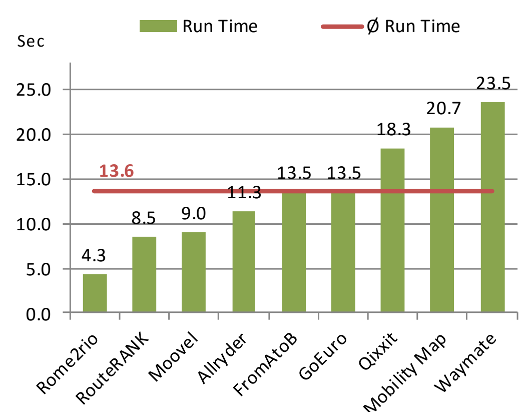


Figure 3. Overview of Run Times.

We have also measured the run time that the platforms needed to process the above requests. The results are shown in Fig. 3. Generally, there is a surprisingly large span between the fastest search at *Rome2rio* and the slowest search conducted by *Waymate*. One of the reasons is that some providers already start computing a possible combination of component services for the most likely origin and destination while travelers are still typing in the corresponding values into the app or on the website. Furthermore, some platforms obviously precompute service combinations for popular origin-destination pairs. Note that there is also a type-related difference in the run time: on average, the construction of a complex service for the local travel request required 10 seconds, and the long-distance travel request required 17 seconds. For the latter, we observed that a flight search engine was included in the search process, which seems to add significant complexity and run time.

To investigate the level of customer-orientation that is provided by today's multimodal mobility platforms, we have analyzed to which extent the mobility service process is supported by each of the platforms. We can state that all platforms offer sufficient support for the *search for information* and *consulting* phases. A choice of complex mobility services is generated from simple spatio-temporal information (when/where), and also types of component services can be selected (e.g. car/no car). It has also become quite common to provide information on the total cost of a service and of service combinations. However, there are limitations regarding the booking and payment phases, which are only supported by three platforms (*Moovel, Mobility Map* and *FromAtoB*). Booking and payment are also limited to selected component services only. The realization phase is only supported by *Qixxit* and *Allryder*. *GoEuro, Rome2rio* and *RouteRANK* also conciliate further, complementary component services such as hotel bookings.

The key to traveler-oriented selection of component services is the processing of detailed service and customer data. However, only five out of the nine investigated mobility service platforms ascertain *static customer data* (such as personal information) at all, and only four of them store them in a customer profile. This goes along with the insufficient support of the booking phase, where customer data is mandatory to finalize a booking. *Dynamic customer data* (such as the current location of a customer) is ascertained by five of nine platforms, and they are mainly used to improve the selection of currently available component services in accordance with the given spatio-temporal characteristics of the traveler.

In sum, only a small choice of service mobility platforms can actually handle a large variety of local as well as long-distance component services and can automatically assemble an appropriate combination of services according to travelers' preferences. In general, the considered traveler and service characteristics remain very simple, and the traveler has only limited control of the selection process. Having selected appropriate component services on a dedicated platform, the traveler usually cannot book the desired complex service, and it is not possible to modify it with hindsight.

3. CUSTOMER-INDUCED ORCHESTRATION

In the following, we embed the traveler-induced combination of mobility services to the customer-induced orchestration of services from several domains. Extending the idea beyond mobility services, customers in general expect improved support of services with respect to the control of selection and bundling today.

In Fig. 4, a selection of relevant domains and corresponding services is shown. For a variety of domains, intuitive apps for smartphones have simplified control of individual services to a great extent. However, apparent weaknesses can still be identified in the combination of component services and in the construction of complex service bundles. This observation does not only hold for mobility services, but also for services in education, finance, and health domains.

As demonstrated for mobility services above, existing platforms lack an intelligent and integrated support of customer preferences, because the control of services is mainly induced by the service provider. Hence, we aim at a *customer-induced control* of services by means of tailored IT platforms. Beyond service selection and bundling, a user centric conduct of services strives for a choice with respect to service providers and aims at the control of complex services during execution.

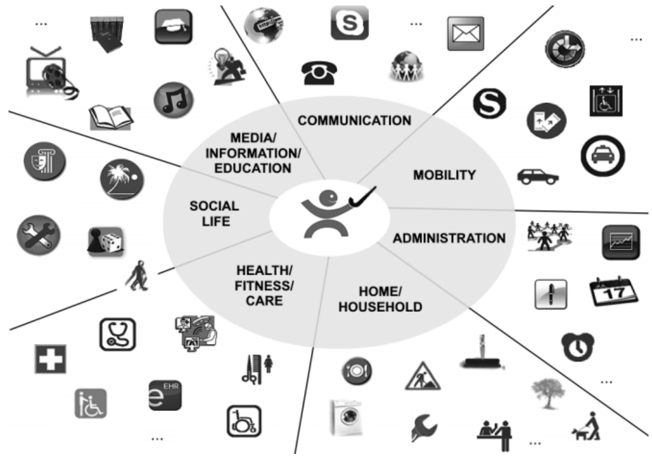


Figure 4. Services for Relevant Application Areas, Adapted from [6]

Abstracting from the mobility service process as shown in Fig. 1, traditionally, a service can be defined as a process of interaction between customer and service provider (Fig. 5). This process is induced by the service provider starting with a *setup of applicable resources*. In the next step, the customer attains *information* about the service from the service provider before the customer and service provider make an agreement and the service is realized (*booking*). The latter phase typically requires the direct interaction of customer and service provider. The process terminates with the *billing* of the service provider and the *payment* of the customer.

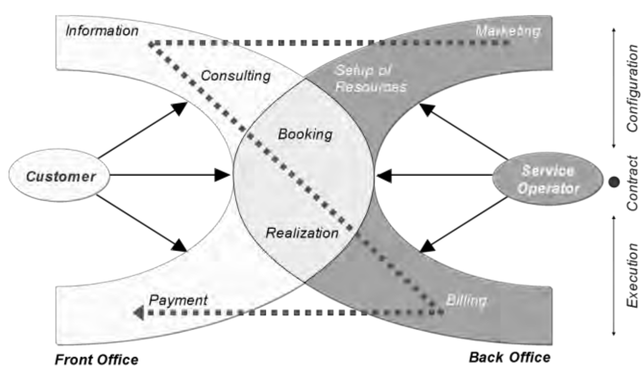


Figure 5. Generic Service Process, following [3].

If several component services need to be combined in order to fulfill the customers' needs, this results in a significant effort of coordination. Typically, a bundling of individual component services into a complex service is required. For each component service involved, the entire service process has to be executed repeatedly. Agents typically take over the control of selecting and combining the component services, hiding the complexity of the complex service from the customer. Today, in the digital economy, the provider of some core (focal) component service often conducts the selection of the remaining services to provide a complex service (e.g. German Railways as focal service on a long-distance trip in the Qixxit platform). As a drawback, the customer gives up control of the details finding him/her confronted with a one-size-fits-all complex service.

This phenomenon can also be observed in the orchestration of so called *smart services*, where control of the component services is achieved by automated service platforms. The black box paradigm of such platforms hinders transparency of service selection and may exacerbate the availability of competing services and/or new innovative component services, though. Furthermore, whenever component services from different service domains have to be combined in order to satisfy some specific customer demand, neither agents nor smart services are available to cope with the complexity of a complex service. Just think of interrupting a business trip in order to see a dentist due to a serious injury. No longer will smart services be available to coordinate the cancellation of a hotel, the re-booking of flights, the correspondence with health insurance and the appointment at a dentist's clinic. Moreover, the cancellation of leisure or sports activities may be involved. The customer himself/herself has to coordinate all combination activities.

The above example incorporates different domains and can be generalized by accepting that customers tend to act in different domains simultaneously. Under this assumption, today's smart services counteract the customer's need to manually control complex services at a detailed component level. Future customer-induced control of service selection and service bundles may alleviate the above sketched weaknesses [2].

To enable customer-induced orchestration, we propose to investigate the following topics using the example of relevant domains such as mobility, finance, education and health and combine the insights in a generic, domain-independent reference model. The core questions for modeling and execution of customer-induced orchestration are:

Modeling of the customer context. Customer-induced orchestration requires information about the situation of the customer, about the customer's preferences and about available component services. The customer context [7] accompanies configuration as well as execution of a component service or a complex service, respectively. While there are several approaches for the modeling of services for individual service operators [8], service operator independent solutions are not widespread yet. Hence, we propose to investigate how the customer context be represented such that suitable component services can be (automatically) selected, combined and configured while personal data is protected from misuse.

Representation of services. There is a semantic gap between the customer's domain language and service operator's domain language, which is a serious obstacle for automated, customer-induced service orchestration. Methods and models are required that present and represent complex and component services appropriately. Hence, we propose to investigate how complex service bundles can be modelled adequately in a domain comprehensive way, and how complex services can be presented to customers.

Methods of automated selection. To enable customer-induced orchestration, component services need to be selected and configured such that they are a good fit with the customer's preferences and such that they fit well together to define a reasonable complex service. Depending on the domain, there are different requirements at methods of automated selection. Hence, we propose to investigate how complex services can be matched with customer profiles and customer contexts, and whether it is possible to incorporate data of former service execution for this task.

Construction of a reference model. Customer context, representation and automated selection need to be condensed by means of a generic reference model which allows the domain-specific as well as domain-independent derivation and implementation of service platforms. However, it is an open issue to which extent domain-specific approaches can be generalized and whether they can be generalized at all. Hence, we propose to investigate whether a reference model does alleviate the above listed issues, and how the degree of user centric control be measured.

4. CONCLUSION

Travelers expect better support in the orchestration of complex mobility services. Mobility platforms promise to select and combine services according to the given preferences of the traveler. Based on an overview of mobility platforms available in German-speaking areas, we have found that the functionality of the existing platforms is rather limited. In particular, these platforms often consider only simple spatio-temporal parameters in the selection of mobility services. Furthermore, the capability of fast intermodal search is often underdeveloped, which leads to long run times and insufficient results of the search.

To ensure customer-oriented combination of mobility services and component services in general, we propose the paradigm of customer-induced orchestration. Our idea is to develop a generic reference model that allows for the conceptualization of mobility service platforms in particular and service platforms in general. A core part of this model is the design of the customer context, a choice of service selection methods such as recommender systems and/or mathematical optimization, and an appropriate representation of services and travelers/customers. We expect that the combination of recommender systems and mathematical optimization will be the methodological core of such a reference model and the derived platforms.

5. REFERENCES

[1] Vogel, P. 2016. *Service Network Design of Bike Sharing Systems: Analysis and Optimization*. Notes in Mobility (LNMOB), Springer.

[2] Winter, A., Alt, R., Ehmke, J. F., Haux, R., Ludwig, W., Mattfeld, D., Oberweis, A., and Paech, B. (2012). Manifest Kundeninduzierte Orchestrierung komplexer Dienstleistungen. *Informatik Spektrum* 35(6), 399-408.

[3] Bodendorf, F. 1999. *Wirtschaftsinformatik im Dienstleistungsbereich.* Springer.

[4] Albrecht, L. and Ehmke, J. F. (2016). Innovative Services in der Mobilitätsbranche: Eine Marktanalyse multimodaler Mobilitätsmanager. In *Proceedings of Multikonferenz Wirtschaftsinformatik 2016*.

[5] Scheer, A.-W. (1998). *ARIS – vom Geschäftsprozess zum Anwendungssystem.* Springer.

[6] Österle, H. and Senger, E. (2011). *Prozessgestaltung und IT: Von der Unternehmens- zur Konsumentensicht*. Controlling & Management, 55, 80-88.

[7] Bazire, M. and Brézillon, P. (2005). Understanding context before using it. In *Modeling and using context*, 29-40, Springer.

[8] Schubert, P. and Koch, M. (2002). The power of personalization: customer collaboration and virtual communities. In *Proceedings of AMCIS 2002*, 269

Exploiting Item Dependencies to Improve Tourist Trip Recommendations

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

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

ABSTRACT

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

CCS Concepts

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

Keywords

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

1. INTRODUCTION AND MOTIVATION

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

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

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

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

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

2. BACKGROUND AND RELATED WORK

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

2.1 Related Tourist Trip Design Problems

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

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

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

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

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

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

2.2 Existing Tourist Trip Applications

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

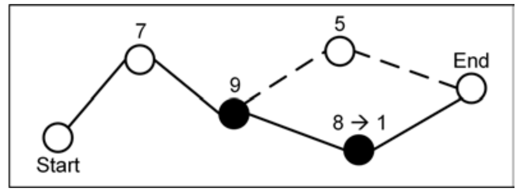


Figure 1: Item dependencies in a tourist trip

2.3 Item Dependencies in Tourist Trip Recommendations

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

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

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

3. PROPOSED SOLUTION AND NEXT RESEARCH STEPS

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

3.1 Extending Existing TTDP Algorithms

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

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

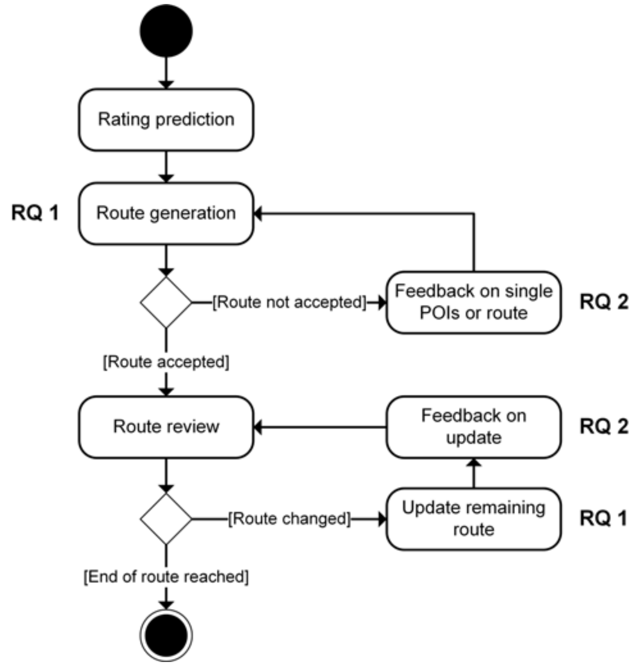


Figure 2: Activity diagram of our framework for generating and updating a tourist trip

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

3.2 Creating Routes and Learning Item Dependencies

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

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

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

3.3 Evaluations and User Studies

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

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

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

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

4. CONCLUSION

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

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

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

5. REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Challenges in Recommender Systems for Tourism

Manoj Reddy Dareddy
University of California, Los Angeles
California, USA
mdareddy@cs.ucla.edu

ABSTRACT
In this position paper, we outline some of the challenges facing recommender systems in the tourism domain. The problems in this domain are unique compared to the traditional recommender systems. The challenges outlined in this paper include: dynamic itinerary planning, mobile platform, evaluation methods, group recommendation, social network, integration, serendipity, user modeling, privacy and robustness. We provide an overview for each of the topics and present the opportunities for improvement. The tourism domain consists of a large amount of information stored digitally and recommender systems can act as a filter that can personalize the experience for every tourist.

Categories and Subject Descriptors
H.4 [**Information Systems Applications**]: Miscellaneous;

Keywords
Tourism; Recommender Systems; Position paper

1. INTRODUCTION
Tourism broadly refers to the movement of people who are exploring new places. Globally, [1] it accounts for 10% of the world's GDP and it supports about 1 in 11 jobs around the globe. It is one of the fastest growing sectors and many nations depend on it as a major source of income. It can be classified into various categories based on their primary motive such as medical, educational, artistic, sports tourism etc. This domain consists of enormous amount of information stored digitally that is not being used to its maximum potential. Recommender systems have huge opportunity in improving the experience of the tourists. This position paper presents various technical challenges that have not yet been addressed by the recommender system community in the tourism domain. The goal of this position paper is to discuss the open problems in this area for researchers to work on.

2. CHALLENGES
2.1 Dynamic Itinerary Planning
One of the main challenges in this domain is optimal itinerary planning for tourists. Tourists generally have an agenda in mind of different places to visit in a city or events to attend, restaurants to try etc. There exist systems that recommend places to visit based on user interest but they are all static in nature. They do not take into account changes that take place in real-time. For example, if a tourist would like to visit Paris, the system should be able to dynamically figure out the opening times and recommend an itinerary. There have been attempts to model this as an optimization problem where the objective function is to maximize a user specific satisfaction metric subject to constraints such as opening times, budget etc. An example of a user specific metric could be the number of places visited or cost etc.

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

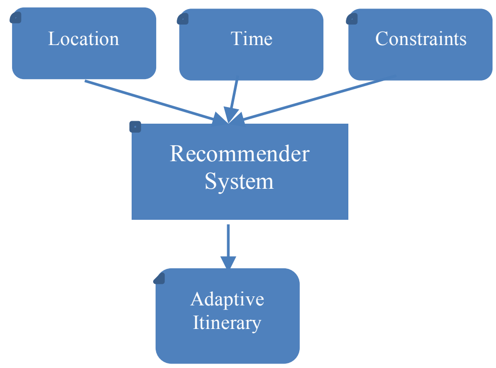


Figure 1: Itinerary generation

An important aspect of such systems is the human interface since it ultimately determines the interaction with the user. In this regard, the design needs to ensure a minimal amount of cognitive effort on the user's part.

2.2 Mobile
The future of computing is mobile. Mobile plays a very important role in this domain since tourists are always on the move. Hence, it is important for recommender systems to take advantage of contextual information such as location, time of day etc. These mobile devices also allow different types of interactions to be captured such as emotion, whether the user is travelling alone or with a group etc. The location information allows the system to recommend events, places to see that are physically close the user.

Figure 2: Mobile devices have a huge role to play in tourism

Another important aspect of mobile that shall play an important role in the future is its ubiquitous nature. This will ensure that the user gets access to the right information at the right time and right location. Current systems such as Google Now, perform such ubiquitous computation by leveraging information from various sources to personalize the user experience.

2.3 Evaluation Methods
The current evaluation methods for recommender systems mostly consider explicit feedback. The most popular techniques being used are Root Mean Squared Error (RMSE) and MAE (Mean Average Error), which relies on explicit user feedback.

$$\sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}$$

RMSE error calculation

Another commonly used evaluation technique is Mean Average Error (MAE) which is defined as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|$$

Mean Average Error calculation

Both these metrics measure the difference between the predicted and actual value on a test dataset. These metrics depend on the explicit user information such as ratings feedback.

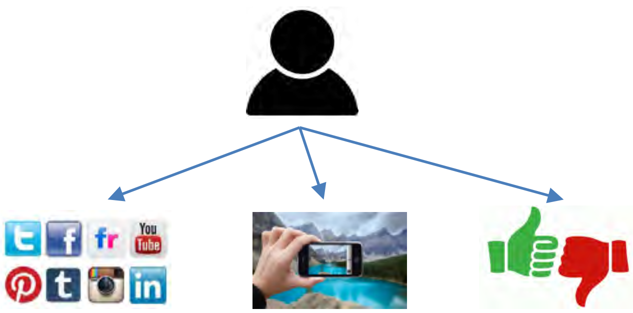


Figure 2: Different forms of feedback by a tourist

Recommender systems in the tourism domain need to be able to gauge user satisfaction level by measuring their emotion in a minimally-intrusive manner. Some of the possible methods includes analyzing the user's social media, pictures being taken and explicit user feedback such as ratings or like/dislike.

2.4 Group Recommendation

Tourists generally travel in groups and current recommendation systems mainly focus on a single user rather than a group. The main challenge is to combine individual preferences of different members and recommend items that are enjoyed by the group as a whole. Certain groups might be more interested in adventure activities whereas others might be inclined towards historical/cultural places. Some of the variables that need to considered are: number of members in the group, individual restrictions and group characteristics.

2.5 Social Network

Social connections play an important role in the recommendation for tourism. For example, if a user's friends recommend trying a restaurant in a different city, then the user is likely to visit that restaurant. There are various types of social influence that ranges from different degrees. One possibility is to integrate existing social network information from sites such as Facebook, Twitter etc. The level of influence depends on the closeness of the user with another user, since it is more natural to trust close friends than users who are 3 or 4 degrees away.

2.6 Integration

The main challenge facing tourists is the integration of various sources of information. For example, the user needs to decide on the airline, hotel, transportation method, tickets to various events etc. It is would be nice to have an end-to-end system that integrates such information in a condense format. The main challenge is to understand the preference of each user and filter out relevant information such as hotel deals etc.

Advertising can play a very important role in recommender systems. A prime example of this is the Google Adwords program. It aims to provide relevant ads that are useful and the user is most likely to click. Similarly, for recommender systems, ads play a very important role since they allow users to learn about relevant promotions such as hotel rooms, restaurant deals etc. Such an interface will allow a tourist to perform all relevant computation without having to switch between different applications which can be cumbersome. Also, it would be helpful if adequate information is provided for various places-of-interest.

Figure 3: Integrated console of all tourist related activity

2.7 Serendipity

Serendipity refers to the idea of discovering a new interest that the user had no idea about. These types of recommendations are the most effective but also the riskiest. The reward is high but the accuracy also tends to be low. In the tourism domain, if a user is interested in art history, the user might be interested in ancient monuments which is a completely different interest. Such models can be learnt using machine learning techniques that process large amounts of behavioral data.

2.8 User Modeling

There also needs to be better user modeling that is able to understand latent user interests. In the tourism domain, the user interests can be organized based on a taxonomy for example: nature, food, etc. This requires building new algorithms that can scale better with different types of input data. Existing techniques such as collaborative filtering, matrix factorization etc. could be applied in this area. Moreover, collaboration with tourism domain experts shall help in better modeling of the user.

2.9 Privacy

Privacy plays a very important role in recommender systems. Since these systems have a lot of personal information, it becomes imperative to protect the privacy of the users. Current systems focus on differential privacy and use aggregates that prevent from identifying individual records.

2.10 Robustness

The systems are vulnerable to manipulation and it becomes important to protect them from various types of attacks. For example, a malicious user might target a competitor by creating fake accounts and down-rating their system, meanwhile increasing the rating of own system.

3. CONCLUSION

The tourism sector presents a number of opportunities for recommender systems. There are many challenges some of which have been outlined in this position paper. These tourism domain specific problems require innovative approaches for implementing recommender systems that can be used by a large number of tourists.

4. ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

5. REFERENCES

[1] World Tourism Organization UNWTO. (n.d.). Retrieved June 23, 2016, from http://www2.unwto.org/content/why-tourism