Proceedings of the Workshop on

Tourism Recommender Systems

The ACM Conference Series on
Recommender Systems

9th ACM Conference on Recommender Systems (RecSys 2015)

Vienna, Austria, September 20th 2015

Workshop organisers:

Dr. Antonio Moreno
Univ. Rovira i Virgili, Tarragona, Spain

Dr. Laura Sebastiá
Univ. Politécnica de Valencia, Valencia, Spain

Dr. Pieter Vansteenwegen
KU Leuven, Leuven, Belgium

Index

Schedule

9:00 – 9:15 **Workshop presentation**

A.Moreno, L.Sebastiá, P.Vansteenwegen

9:15 - 10:30 ***Invited talk: The many roles of knowledge in tourism recommender systems (Markus Zanker, Alpen-Adria-Universität Klagenfurt, Austria).***

When tourism as a social phenomenon meets the technical application domain of recommender systems a myriad of challenges arises. This talk explores how domain understanding and knowledge-based methods contributed to the success of a diverse range of applications and discusses their limitations and avenues for future research.

10:30 – 11:00 **Coffee break**

11:00 – 12:30 ***Session 1: Semantic Tourism Recommender Systems***

A highly interactive tourism recommender system for multi-day trips
L.Sebastiá, D.Yuste, I.García, A.Garrido, E.Onaindia

Semantic diversification of touristic recommendations
J.Borrás, A.Moreno, A.Valls

The Next Thing - Connecting the museum visit
A. Lo Bue, A.J.Wecker, T.Kuflik, A.Machi, O.Stock

12:30 – 14:00 **Lunch**

14:00 – 15:30 ***Demo session***

Design and development of a real-time RecSys based on location, mobile device and tourists activities

S.A.Jazdarreh, E. de Quincey, I.Mackinnon

Visit Costa Daurada & Terres de l'Ebre: a semantic recommender system of tourist activities

J.Borrás, A.Moreno, A.Valls

Finding more fun for travelers

S.Donohue, J.Heath, S.Maclachlan

15:30 – 16:00 **Coffee break**

16:00 - 17:30 ***Session 2: Advanced topics in tourism recommender systems***

CT-Planner5: a computer-aided tour planning service which profits both tourists and destinations

Y.Kurata, Y.Shinagawa, T.Hara

The importance of diversity in profile-based recommendations: a case study in tourism

F.Sánchez-Vilas, J.Ismoilov, E.Sánchez

TravelWithFriends: a hybrid group recommender system for travel destinations

T. De Pessemier, J.Dhondt, K.Vanhecke, L.Martens

CLG-REJA: A consensus location-aware group recommender system for restaurants

J.Castro, O.Cordón, L.Martínez

A highly interactive tourism recommender system for multi-day trips

Laura Sebastia
Univ. Politecnica de Valencia
Camino de Vera s/n
Valencia, Spain
lstarin@dsic.upv.es

David Yuste
Troovel.com
San Vicente, 150
Valencia, Spain
david.yuste@troovel.com

Inma Garcia
Univ. Politecnica de Valencia
Camino de Vera s/n
Valencia, Spain
ingarcia@dsic.upv.es

Antonio Garrido
Univ. Politecnica de Valencia
Camino de Vera s/n
Valencia, Spain
agarridot@dsic.upv.es

Eva Onaindia
Univ. Politecnica de Valencia
Camino de Vera s/n
Valencia, Spain
onaindia@dsic.upv.es

ABSTRACT

This paper presents `troovel.com`, a Tourism Recommender System (TRS) that helps the user plan a multi-day trip anywhere in the world. The system presents various precomputed agendas that the user can adapt to her own requirements and preferences. `troovel.com` records the interaction of the user while navigating the web site and uses these data to offer personalized recommendations. Recommendations are calculated by applying a content-based technique and a novel filtering approach based on dynamic relationships between items extracted during the user navigation. Results show that users are more responsive to recommendations provided by our TRS than those offered by a standard recommendation.

1. INTRODUCTION

Tourism Recommender Systems (TRSs) aim to match the user preferences with the leisure resources and tourist activities of a city [10]. TRS need some initial data, usually explicitly provided by the user, and they are especially useful if they can automatically infer the user preferences through an explicit or implicit feedback. Most TRS are hybrid approaches that combine basic recommendation techniques such as demographic (DM), content-based (CB) and collaborative filtering (CF) techniques. An analysis presented in [1] shows that more than half of the analyzed works (53%) use a mixture of these three techniques.

Also importantly in TRS is the capacity of providing a personal agenda to the user with the set of recommended tourist activities. Some approaches even offer a personalized timetable taking into account the context information such as the opening and closing time of the attractions, the time needed to go from one place or another or even the transportation means. *e-tourism* [3], for instance, is a tourist recommendation and planning application to assist users on the organization of a leisure and tourist agenda. First, a recommender system offers the user a list of the attractions that are likely of interest to the user. The recommendation technique applies a hybrid approach that combines the demographic classification of the user (DM), the user likes in former trips (CB) and the preferences for the current visit. Second, a planning module schedules the list of recommended places according to their temporal characteristics as well as the user restrictions. Hence, the planning system determines how and when to realize the recommended activities, according to distances between places, opening and closing hours, etc.

The same idea is followed by *CRUZAR* [7], a web application that builds custom tourism routes for each visitor in the city of Zaragoza (Spain). *CRUZAR* provides a web form where the user profile is captured, essentially her preferences and the trip context, and applies a mixture of DM and CB recommendation techniques to elicit the most suitable attractions for the user. At the last stage, the system generates a customized route for the visitor, taking into account the distances between the attractions, the subjective interest of tourism resources, previously calculated, the particular circumstances of the trip (number of visitors in the tour, dates of the trip) and other relevant aspects such as opening and closing time of museums and churches.

Otium [8] and *City Trip Planner* [11] show a scheduled route of attractions for one day (selected by a CB algorithm) represented through a timetable and a map. They also allow the user to interact with the presented plan to add or remove activities, change the order of visit, etc. Moreover, they enable to download the route to a mobile device so that the tourist can follow the plan during the trip. *CT-Planner* [4] [5] offers tour plans in several areas of Japan which are progressively refined as the user indicates her preferences and requests (duration, walking speed, reluctance to walk). The recommendation is performed by a CB algorithm and the planning problem is formulated as a *Selective Travelling Salesman Problem* [6]. *SAMAP* [2] is a case-based reasoning software tool that elicits a tourist plan given the past experience of the system with similar users. It provides indications

TouRS15 Workshop RecSys 2015

about the transportation means, including walking; recommends restaurants and bars for lunch or dinner, and suggests leisure attractions such as cinemas or theaters. *SAMAP* allows the user to download a file with the geo-referenced map with a detailed explanation of the plan.

Regarding the user interfaces of TRS, most of them offer a web-based interface and/or an interface specifically designed to be used in mobile devices. The work in [1] presents an excellent review of user interfaces in recent TRSs. Authors conclude that although web-based interface is the most commonly used choice for its ease of access and use, more than half of the reviewed systems have specific interfaces for mobile devices. One example is *GeOasis* [9], a tourist guide that shows the description of the attractions as the tourist approaches the recommended locations, thanks to the GPS functionality of the mobile device. *GeOasis* predicts the immediate future user location, e.g. in a city, near a city or on the road, and the algorithm selects the most relevant attractions. The plan is computed by a client application.

This paper presents `troovel.com`, a TRS that helps the user plan a multi-day trip anywhere in the world. The system presents various pre-computed agendas that the user can adapt to her own requirements and preferences by including and removing places to visit or moving one visit from one day to another. `troovel.com` captures and records the interaction of the user in the current session and uses these data to offer recommendations, thus avoiding having to request the user to fill in any prior questionnaire. Recommendations are calculated by applying a CB technique and a novel CF approach, based on dynamic relationships between items, extracted from the interaction of the user with the system.

The paper is organized as follows. Next section describes the ontology to classify the recommendation items as well as the database that stores information about the users and items. Section 3 explains all the components and developed techniques of the TRS. Section 4 outlines the user interaction with the system, how recommendations are obtained and it provides some implementation details of the user interface. Section 5 shows a simple example of application. Section 6 analyzes and evaluates the information extracted from the user interaction with the system. We finish with some conclusions and further work.

2. ONTOLOGY AND DATA

The ontology used in our approach describes the most important features to classify and label the items (i.e. touristic places). It is created as a very expressive hierarchy of categories, extended with information from TripAdvisor (`http:/www.tripadvisor.com`) and OpenStreetMap (OSM, `http://www.openstreetmap.org`) that allows us to label and group items with similar characteristics and, therefore, match and recommend items according to the users preferences. A small fragment of the ontology is depicted in Figure 1. As a whole, it comprises nearly 100 labels which are classified into six wide categories (including leisure, architecture and monument, point of interest, natural, route, and museum) that are subsequently classified up to three subcategory levels to consider more detailed aspects, such as Leisure|Show|Ballet, Leisure|Shopping|Market, Natural|Water|Lake or Museum|Planetary among others.

The underlying idea is to classify the items into one or more (sub)categories. In order to have our items better labeled and to further improve the quality of the final recommendation, each item is not only classified in terms of true/false values but they are associated to an adequacy ratio of the item to the category, meaning that a given item may fit slightly better in one category than in others. For example, the Big Ben in London may be labeled with an adequacy ratio of 90% in Architecture and monument|Civil building with architectural value category, and 100% in Point of interest|Attraction. These ratios have been automatically initialized but can be tuned by data mining techniques.

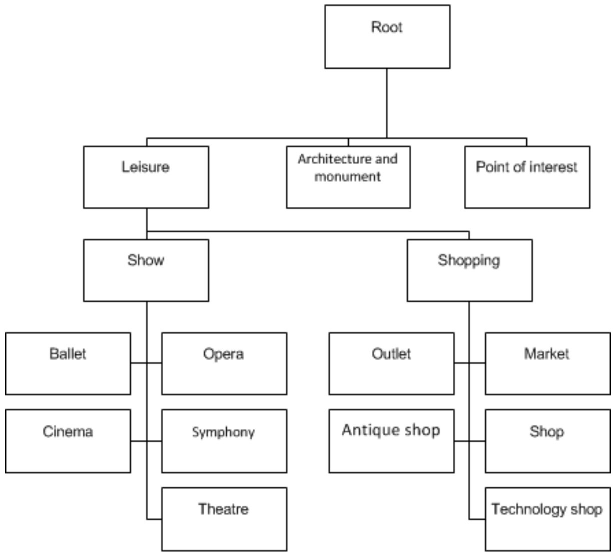


Figure 1: A fragment of the ontology used in our approach with the hierarchy of categories.

As a data source we use a mySQL database with more than 150 relational tables. Although a complete and technical description of these tables is beyond the scope of this paper, for the sake of simplicity we will group these tables into three blocks. The first block comprises all the compiled information about the items. This involves a high number of tables to store a lot of information about the places associated to items[1], which includes the typical identification data such as name, description, geographical coordinates, a rating to denote the general interest of this place, ISO codes to easily find the countries they belong to, multimedia information (mainly pictures and videos), the type of item (point of interest, amenity, shopping, etc.), and *autocompletion* information with some mapping and dictionaries to deal with search in different languages, typically Spanish and English. This information has been automatically populated from TripAdvisor, OSM and Wikipedia sources. Obviously, every item is labeled and categorized according to the ontology described above and annotated with extra tags and information, such as opinions and comments, retrieved from social networks like Facebook, Google, Twitter, Flickr and Wikipedia.

To this end, first raw data from external sources is fetched. Web scraping and direct web services are used. Then data is processed by scripts that extract the relevant information:

[1]We currently have almost 2 millions items, but we still plan to increase this number in the mid-term future

place name, geolocation, tags and attributes of the places, user ratings, etc. We only look for POIs, since the characteristics of cities are assumed to be related to their POIs.

In order to be able to manage external data, external places must first be mapped with the ones listed in our data base. An algorithm combining geographical distance, Levenshtein distance on names, and additional heuristics determines if two places are the same one. The POIs that do not match anyone in our database are inserted as new ones.

Once places on every external source are related to our database, semantic information is considered. Mapping rules specialized on every data source establish relations among external information and our ontology (eg. TripAdvisor tags, OSM classification, etc.). POIs may be classified on different ontology classes, and weighted depending on normalized external user ratings. At the end of the process, the classes of POIs are propagated up to their parents (ie. cities), by combining and normalizing again their weighted ontology classes attributions.

The second block contains the information about the users, both registered and anonymous users. In both cases we capture the information of the navigation events while the user interacts with the trips recommended by the TRS; this implies monitoring and storing the events generated when a user makes an explicit search of an item, clicks on an item in the map for extra information or just enters a page to view the items without any further action. The registered navigation events gives us some hints about how appealing (and successful) an item can be for future users queries, thus helping us improve the importance ratio of such an item in our database.

The information gathered during the navigation events is extended to capture the explicit interest of the user in the items of the stored trip (in the case of registered users) or in the items of the trip in the current session (in the case of anonymous users). This new information is collected whenever the user adds/removes items to/from the trip, or simply marks an item as *already visited*, discarding it from this trip, and probably other future ones in case of registered users. In addition to the personal data, we also store the user preferences (likes, dislikes and wished items and categories) in terms of our ontology. This is the essential input information the TRS needs to return highly personalized trips. Finally, we also store the comments of users on the items and trips —we have additional relational tables to support the moderators' tasks.

The third block involves the tables to store the output information provided by the recommender system. From an advisory point of view, this includes the intelligent recommendations given by the system to fit the user preferences individually. In short, the recommendation consists of a guided trip with the route to follow and the most interesting items to visit along the route. How this recommendation is calculated and the used techniques are explained in the next section.

3. RECOMMENDER SYSTEM

The main purpose of our TRS is to create a personalization tool that recommends a person a list of information items that best fit her individual tastes. A recommender system infers the user preferences by analyzing the available user data, information about other users and information about the environment. The adequacy of the recommendations highly depends on the amount of available information.

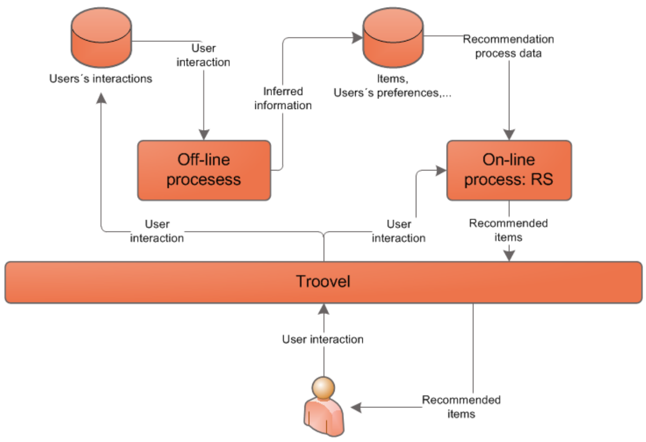


Figure 2: Recommender system.

The system is integrated into a web service and so recommendations must be calculated in real time. For this reason, the most time-consuming processes are executed as batch processes, i.e. in an off-line manner, when possible.

We have created two types of processes, which can be invoked independently (Figure 2):

1. Off-line processes:
   - Updating the user preferences. This process is necessary for the CB recommender system.
   - Extracting dynamic relationships between items to infer statistical correlations and, possibly, dependencies between items. This process is necessary for the *item2item* recommender system.
2. On-line process: this is the recommendation process itself, which elicits a list of recommended items adapted to the user who is currently using the system.

Both types of processes can be run independently of each other. In general, the more updated data calculated by the off-line processes, the better the recommendation of the on-line process. The two off-line processes are also independent and they are executed when deemed appropriate. Preferences are used for the CB recommendation technique and the relationships between items are used for the *item2item* recommender system. Therefore, the more accurate the data, the more reliable the recommendation. However, if the two off-line processes are unnecessarily run, a lot of time will be wasted. The system must thus decide when to run the off-line processes to improve the recommendation while avoiding a system overload.

3.1 Off-line processes

3.1.1 Updating preferences

This is a batch process that updates the user preferences taking into account the interaction of the user with the system. The user preferences is a list $P_u = \{(p, r_{up})/p \in P, r_{up} \in [0..100]\}$, where, P is the set of categories in the ontology and r_{up} is the affinity of the user with the category p. P_u is updated whenever a recommendation is calculated in order to maintain P_u is as accurate as possible.

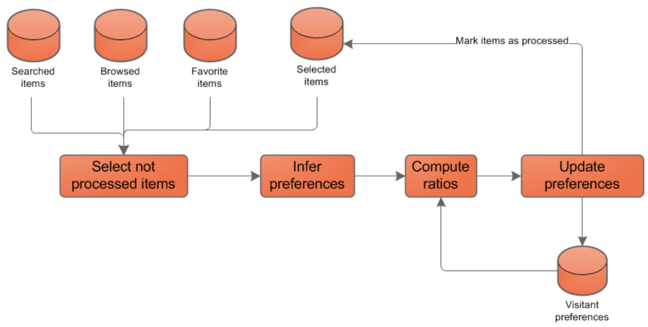


Figure 3: Updating preferences process.

The purpose of this process (Figure 3) is to update the ratios of all preferences related to the items with which the user interacted some way. The set of items I which the user has interacted with are: items searched and items browsed by the user, the user favorite items and the items that the user has explicitly selected. The items used in this process are added to a list named L_{up} and marked as "processed" so as not to be reused in subsequent updating processes.

The steps of the updating process are:

1. Create the list L_{up}, where, for each item in L_{up}, the interest ratio of the user on the item is estimated. Each item of the list contains the following information: $L_{up} = \{(i, r_s, r_n, r_f, r_{sel})/i \in I, r_s \in [0..100], r_n \in [0..100], r_f \in [0..100], r_{sel} \in [0..100]\}$, where r_s is the ratio of the item considering the search made by the user, r_n is the ratio considering when the user has accessed that element as a result of the navigation, r_f is the ratio considering that the user has marked the item as favorite and, finally, r_{sel} is the ratio considering that the user has explicitly selected the item.

2. For each item in L_{up}, we retrieve the categories under which the item is classified in the ontology (user preferences). These preferences are included in a list IP_{up}, where $IP_{up} = \{(p, r_p)/p \in P, r_p \in [0..100]\}$. The interest degree r_p is computed using the ratios of the items in L_{up} (r_s, r_n, r_f and r_{sel}) that are classified in the ontology under the category (preference) p.

3. Update the ratios of the user preferences in P_u using the ratios of the same preferences in IP_{up}. The new ratio will be the average of the new ratio and the old one. If a particular preference is not present in P_u, it is added with the new ratio. Thus, the ratios of the preferences are increasing or decreasing to fit new tastes of the user.

Hence, next time a recommendation is requested, the new user preferences will be used.

3.1.2 *Extraction of dynamic relationships between items*

In order to improve the quality of the items used for the recommendation, we implemented a batch process to extract dynamic relationships between items. The underlying idea is that two items will be more or less related whether or not they appear together throughout the navigation events of the users. This way, we can infer a correlation between items. Intuitively speaking, if one item appears during the user navigation, the frequency of appearance of the other item during the same navigation session provides extra advice to the recommender as a very simple, but effective way to achieve a kind of *serendipity*.

Given a list L with all the items of the data source, the steps for this process are:

1. Initialize $L_{i1,i2}$ as an empty list of pairs of items $\langle i1, i2\rangle$ that are correlated.

2. For each item $i1 \in L$, count the number of times, namely $total_{i1}$, that $i1$ appears in the list of navigation events (item visited, wished, searched, clicked, etc.)

   (a) For each item $i2 \in L, i2 \neq i1$, count the number of times, namely $total_{i1,i2}$, that $i1$ and $i2$ appear together in the list of navigation events. In other words, the number of times $i2$ appears, subject to $i1$ appearing as well.

   (b) Define $freq_{i1,i2} = total_{i1,i2}/total_{i1}$ as the ratio of $i1, i2$ appearing together divided by the number of times that $i1$ appears.

   (c) If $freq_{i1,i2}$ is greater than a given threshold, insert $\langle i1, i2, freq_{i1,i2}\rangle$ into $L_{i1,i2}$. In our implementation we consider threshold=0, but we can require a stronger or weaker correlation index by simply tuning the value of this threshold.

Let us assume three items {*Big Ben (BB), London Eye (LE), London Aquarium (LA)*} which appear 20, 15 and 10 times, respectively, in the list of navigation events. *BB* and *LE* appear together 8 times, whereas *BB* and *LA* appear together 4 times. This way, the correlation value for *LE* subject to *BB* is 8/20=0.4, and for *LA* subject to *BB* is 4/20=0.2. This means that *BB-LE* are more correlated than *BB-LA* and, obviously, when *BB* is recommended the system will recommend *LE* with a higher probability than *LA*.

Extracting these correlation relationships is a very simple but time-consuming task since we need to analyze every pair of items to calculate their *similarity* value in the list of navigation events. Therefore, this task is only executed when the system overall load is low. Nevertheless, this task does not affect negatively the global performance of the system nor the recommendation process.

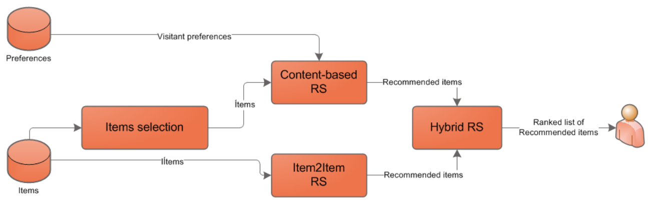


Figure 4: Recommendation process.

3.2 On-line process: the recommendation process

The RS (Figure 4) combines two basic recommendation techniques (item2item and content-based) into a hybrid RS thus alleviating the limitations of using one technique over the other.

The recommendation process starts by calling the method *getRecommendation* with the user identifier u, the geographical area where the recommended items should be located, the type of items to recommend (places, points of interest,...), and the maximum number of recommended items to be returned as a result of the recommendation process (*limit*).

The output of the recommendation process is a list of ranked items LRI_u. The items are ordered according to the estimated interest degree of the user in the item. The items are included in the geographical area and are classified into the requested categories. Each item in the list consists of the item identifier and the computed item ratio.

The recommendation process steps are:

1. Select the items that may be recommended. That is, select the items located in the geographical area that belong to the requested categories. The selected items are included in a list L. Items in L contain the following information:
   - Item identifier *id*.
   - The ratio r_{cb} computed by the CB recommendation; this is the degree of affinity of the user in the item according to the content-based RS.
   - The ratio r_{i2i} computed by the item2item recommendation; this is the degree of affinity of the user in the item according to the item2item RS.
   - The ratio r_h computed by the hybrid RS; this is the degree of affinity of user in the item according to the hybrid RS.
2. The content-based RS calculates r_{cb} for each item in the list L. r_{cb} will be 0 if the RS considers that the item is not of interest for the user.
3. The item2item RS calculates r_{i2i} for each item in the list L. The ratio may also be 0.
4. The hybrid RS combines r_{cb} and the r_{i2i} and obtains r_h for each item in L. Afterwards, the RS sorts the list L by r_h.

3.2.1 Content-based recommendation

The CB recommendation process has been simplified as far as possible to make it fast. Updating and obtaining the user profile is performed in an off-line process (see Section 3.1.1).

The RS uses only the data of the user currently logged on (user u) for recommendation; that is, the user preferences $P_u = \{(p, r_{up})/p \in P, r_{up} \in [0..100]\}$, where P is the set of categories in the ontology and r_{up} is the affinity of the user with the category p. P_u must be updated so that when the recommendation is calculated, P_u is as accurate as possible.

The result of the CB recommendation process is the list L, where the content-based ratio r_{cb} of each item in the list has been conveniently updated. The CB recommender system uses P_u and the classification of the item in the ontology to compute r_{cb}.

3.2.2 Item-2-Item recommendation

Item2Item is a novel recommendation technique based on finding relationships between items that are not directly related. This technique finds surprising relationships between items. The relationships between items are obtained using the off-line process described in Section 3.1.2). The result of this process is the list L, where the item2item ratio, r_{i2i}, of each item in the list L, is conveniently updated.

The off-line process produces a list LRI of related items: $LRI = \{i_1, i_2, total_{i_1}, freq_{i_1,i_2}, r_{i_1,i_2}\}$, where $total_{i_1}$ is the number of occurrences of i_1, $freq_{i_1,i_2}$ is the number of joint occurrences of i_1 and i_2 and r_{i_1,i_2} is a value computed as $freq_{i_1,i_2}/total_{i_1}$ and weighted by the context of the relationship.

The item2item recommendation process has the following steps:

1. Get a list LRI_u (a subset of LRI) containing only the tuples where i_1 is related to the user. The items related to the user are the ones the user has selected on previous interactions with the system. $LRI_u = \{i_2, total_{i_1}, freq_{i_1,i_2}, r_{i_1,i_2}\}$.
2. For each item i_2 in the list LRI_u:
   (a) Get all occurrences of i_2 in the list LRI_u. Compute the ratio r_{i2i} taking account of the multiple item occurrences and the values of $total_{i_1}$, $freq_{i_1,i_2}$, r_{i_1,i_2} of each one.
   (b) Update the list L with the r_{i2i}. If the item i_2 is not in L, it will be added.

3.2.3 Hybrid recommendation

As we mentioned before, basic recommendation techniques exhibit some disadvantages that are alleviated by using an hybrid RS and combining their results. Specifically, our hybrid RS combines the content-based recommendations with the item2item recommendations.

First, the hybrid RS computes the hybrid ratio for each item in the list L, r_h, using the basic RS ratios (r_{cb} and r_{i2i}) and the ratio in the database that refers to the general interest of the user in this item. Second, the RS orders the list according to the ratio r_h (from highest to lowest). And finally, it creates the recommended item list.

The recommended item list $LRSI_u$ contains the items of L with the highest ratio (the number of items is defined when invoking the recommendation process through the parameter *limit*). Each item in the list is associated a tuple with the item identifier and the ratio r_h calculated by the hybrid RS.

4. USER INTERACTION

In this section we introduce the platform where our recommender system is deployed. We focus on user perception and interaction, and how information is retrieved from user behavior in order to feed our learning algorithms.

Users can interact with our recommender system through the web site `http://troovel.com`. It is a state of the art JavaScript application that connects to the artificial intelligence system and all the miscellaneous subsystems asynchronously via AJAX. The first interaction with the application shows a selection of geolocalized trips over a map, as in Figure 5. The user may look for destinations by text search or explore the map looking for more trips.

After looking for a destination, if none of the proposed trips satisfies the user, the application lets him create a blank trip for the searched destination. Otherwise, if an interesting trip is found, it can be joined by clicking on it. In both cases (blank and proposed trips), the user gets into the trip detail

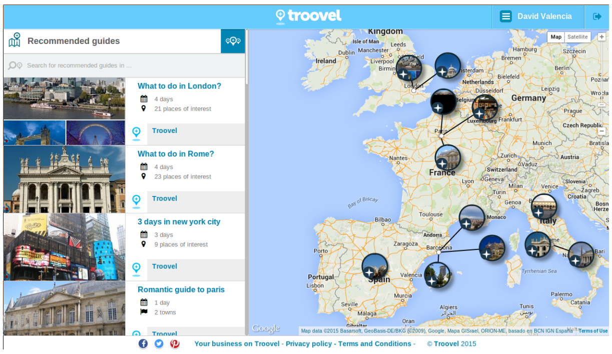


Figure 5: First contact with the application: a list of predefined trips.

screen (Figure 6). On the left panel, the itinerary of the trip is showed: it is a sequence of days and places that should be visited these days. On the right side, the map shows the same places and the route among them.

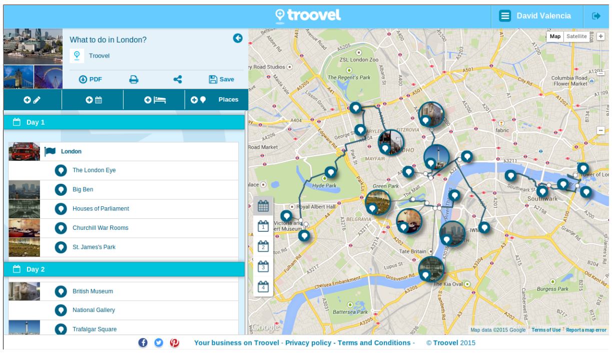


Figure 6: Detail of a trip.

The user may add new places to the trip by clicking on the *add place* button. A list of places shows up on the left panel and over the map. By exploring the map new places are proposed. Places can also be searched by writing its name on the left panel.

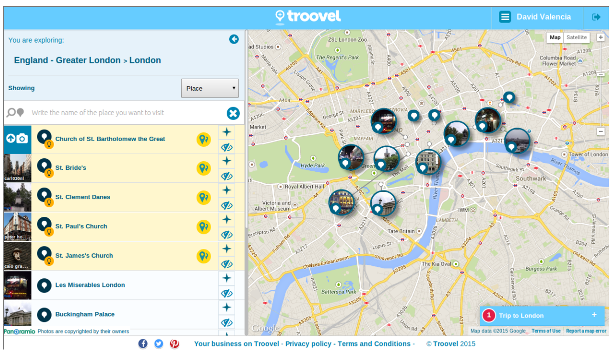


Figure 7: List of proposed places.

Figure 7 shows the list of proposed places. Pay attention to the first five highlighted places. These places have been recommended by our algorithm. Recommended places have an explicative icon on the right side, briefly describing the main recommendation reason. In the example the top five places arise because of content based technique, thus we expose to the user that the characteristics of these places matches his tastes, which have been previously learned by the system. By clicking on the explanation icon, the screen of Figure 8 appears, with detailed recommendation reasons.

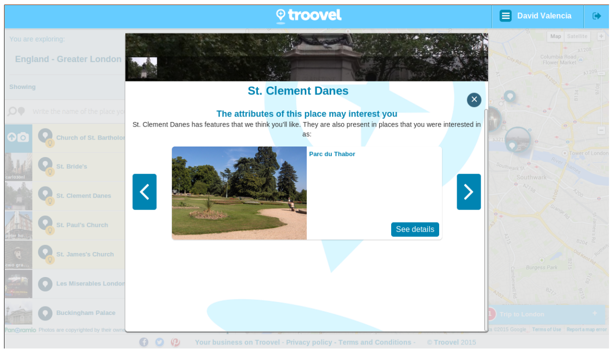


Figure 8: Explanation of a recommendation.

Besides a textual explanation of the reasons for this recommendation, the list of influential places in which the user was previously interested and contributed to the recommendation is shown. Three possible explanations are displayed and influential places as well as people are listed below each explanation. In order to retrieve this information, the application collects events each time the user clicks on a place, looks for it in the searcher box, adds a place to the trip, or tells us the user was there. All these events are sent to the web server and they are queued in the data base.

A C++ daemon linked to the libraries implementing our TRS dequeues user events and performs learning. Different calculations are done with distinct frequencies, depending on their computational cost. Every ten seconds single user preferences are updated (see Section 3.1.1). The current recommendation along with its reasons are fetched from a C++ server listening for HTTP requests. It is a boost based server, with 32 threads dedicated to web services. The TRS requests are served in an average of 33 milliseconds, as shown in Figure 9 and Figure 10.

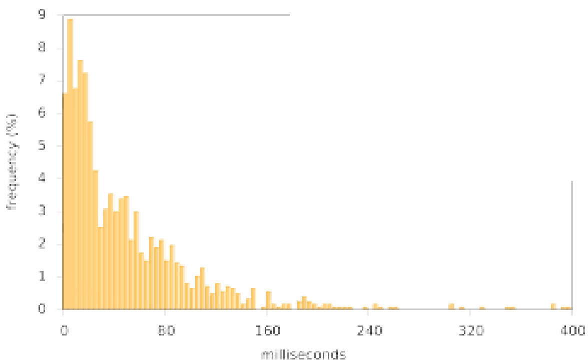


Figure 9: System response time.

5. EXAMPLE OF APPLICATION

In a simplified scenario, a user arrives to the website and starts to customize a trip to Valencia. The user is a fresh new

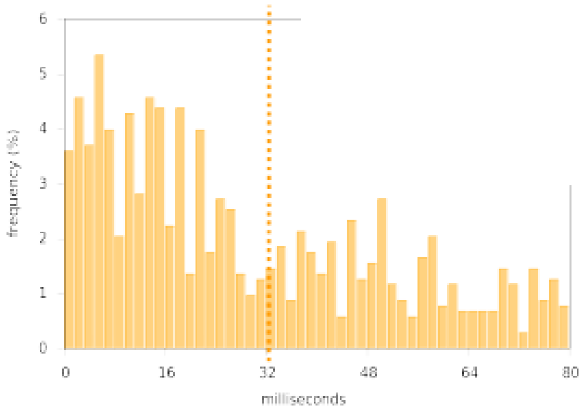


Figure 10: Detail of average system response time.

one, so there is no knowledge about his preferences on the database. Thus a popular set of places is proposed. Among them, the user pays attention and click the three POIs in Table 1.

Table 1: POIs clicked by the user

POI	Ontology classification	Weight
Miguelete Tower	architecture and monument / tower	45
Valencia Cathedral	architecture and monument / worship building / cathedral	61
Round Square	architecture and monument / civil building with architectural value	32

The POIs are classified and weighted as shown in Table 1. The interface sends information of each click to the server, which registers the event. The *updating preferences* learning batch process detects new user interactions pending to evaluate, and user profile is updated. The user is linked to the classes of the ontology related to the clicked POIs (and since it was the first user interaction, the same weights are applied to the relations) as shown in Table 2.

Table 2: User ontology relations

Ontology class	Weight on user preferences
architecture and monument	46
architecture and monument / tower	45
architecture and monument / worship building	61
architecture and monument / worship building / cathedral	61
architecture and monument / civil building with architectural value	32

When a subsequent recommendation occurs, AI results arise. A set of POIs in the context of Valencia are evaluated, and each recommender rates it. The most relevant places with architecture and monument class become relevant because of content-based recommender. Some of the best rated results are listed at Table 3.

Table 3: Most relevant POIs for the user

POI	Ontology classes	Weight
North Station	architecture and monument / civil building with archit. value	48
St. Agustin Church	architecture and monument / worship building	40
Quart Towers	architecture and monument / civil building with archit. value + defensive architecture	38

Also Item2Item recommendation arises because of Round Square POI, which has been related to The Lonja due to previous interaction of different users. It turns to be another *architecture and monument* POI, so after hybridation it end ups with a high enough rating to be included in the final recommendation. As the user interaction continues, more information from the user allows more accurate recommendation with a better balance between the different recommendation techniques.

6. ANALYSIS OF THE USER INTERACTIONS

This section analyzes the information captured while the user interacts with the system. We also analyze the geographical area and the ontology category of the items that have been recommended. A log with the users interactions was recorded during 3 months. This log stores the following events:

- Recommendations:
  - List of items received by the user as a popular recommendation
  - List of items received by the user as a result of our hybrid RS
- Feedback:
  - Items the user has clicked on to see more details
  - Items added by the user to the trip
  - Items marked as already visited by the user
  - Items the user dislikes

We consider the three first feedback actions as positive feedback. On the other hand, the low activity in the system is reflected in the low amount of available information. Specifically:

#unique users	2182
#users that have received only popular recommendation	2131
#users that have also received recommendations from the hybrid RS	51
#users that have performed any action on popular recommendations	220
#users that have performed any action on recommendations from the hybrid RS	18

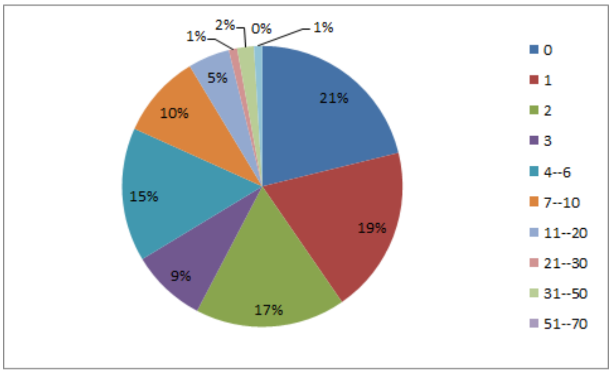


Figure 11: Percentage of positive interactions of each user wrt popular recommendations.

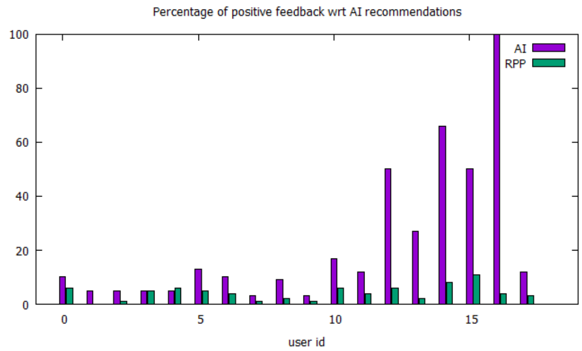


Figure 12: Percentage of positive interactions of each user wrt AI recommendations.

Since a certain interaction with the system is necessary for obtaining results from the hybrid RS, the difference in the number of users who performed feedback actions on popular recommendations and on recommendations returned by the hybrid RS is expected. However, the feedback over hybrid RS recommendations indicates that, even though the number of analyzed users is low, the personalized recommendations are more accurate. Specifically, the total number of items recommended by popular recommendation are 130189, where only 1225 received positive feedback, that is, 0.94%.

On the other hand, the total number of items recommended by the hybrid RS decrease to 1212, but 81 out of the total received positive feedback, which implies 6.68%. This is detailed in Figures 11 and 12, which show the percentage of positive feedback that users gave to the items recommended by the popular and by the hybrid recommendation, respectively. Figure 11 shows the percentage of users whose number of interactions falls in the corresponding range. This chart indicates that the majority of users have provided feedback for less than the 3% of the recommended items. In fact, the average of percentage of feedback is 4.46 and the standard deviation is 10.13. In Figure 12 two columns are shown for each user, indicating the mentioned percentage for hybrid and popular recommended items, respectively. In this case it is clear that, for this set of users who showed a more highly interactive session with the system, the feedback they provided is much more positive for items recommended by the hybrid RS than for popular recommendations. Therefore, we can conclude that, although the interaction of the users with the system is low up to now, the results obtained by the hybrid RS are promising.

Table 4: Coverage of the catalog of items classified in the categories of the first level in the ontology hierarchy and non-classified recommendation items

Category	Number of recommended items	Number of items in the category
architecture and monument	4277	19651
leisure	7229	48354
museum	3105	15253
natural	2325	20498
point of interest	3440	16461
route	236	2098
		Number of total recommended items
non-classified	34966	49995

Figure 13 shows a map[2] with the location of the items recommended by both the popular recommendation and the hybrid RS. The map shows that the majority of requested recommendations are within Europe, coast in both North and South America, India and Japan.

Table 4 shows the number of recommended items that belong to the categories of the first level in the ontology hierarchy. These items are recommended by the content-based technique. The last row of this table shows the recommended items that are not classified in the ontology. This indicates that the *item2item* technique was able to recommend many items thanks to the computation of the relationships between these items.

7. CONCLUSIONS

In this paper we have presented `troovel.com`, a TRS to plan a multi-day trip in any place around the world. `troovel.com` is a highly interactive tool that allows the user to adapt the trip agenda to her preferences and it also records the navigation session of the user in order to learn her likes and preferences. Our TRS uses a content-based recommendation technique and the *item2item* technique, a novel method based on the dynamic relationships between items captured during the navigation session of the user. Using this hybrid recommendation technique alleviates the shortcomings of using only basic techniques. Specifically, while the CB technique captures the general likes of the user based on her past experience, the *item2item* technique is very appropriate to capture the user intentions at the particular time of the navigation session.

The results in Section 6 show that although we do not dispose of a large collection of data, users are responsive to the recommendations provided by the hybrid RS, thus indicating a greater satisfaction than with items returned by a classical popular recommendation.

[2]CartoDB attribution ©OpenStreetMap contributors ©CartoDB

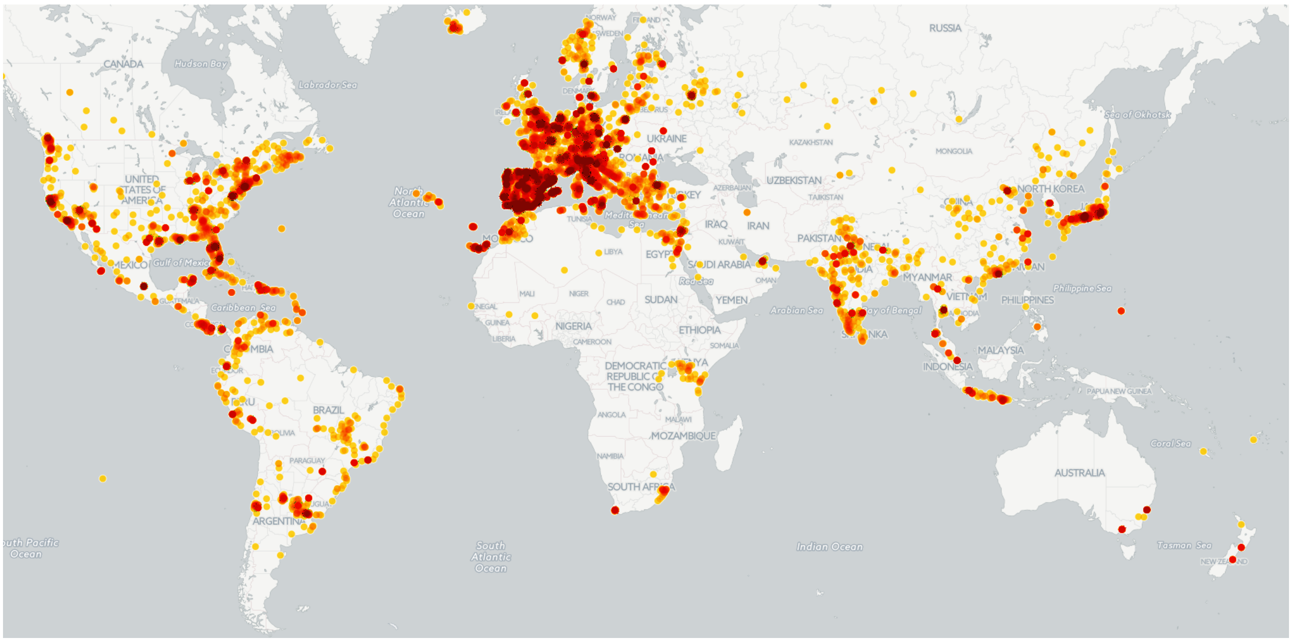


Figure 13: Spatial distribution of recommended items.

8. ACKNOWLEDGMENTS

This work has been partly supported by the Spanish project TIN2014-55637-C2-2-R and the Valencian project PROMETEOII/2013/019.

9. REFERENCES

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

[2] L. A. Castillo, E. Armengol, E. Onaindia, L. Sebastia, J. González-Boticario, A. Rodríguez, S. Fernández, J. D. Arias, and D. Borrajo. samap: An user-oriented adaptive system for planning tourist visits. *Expert Syst. Appl.*, 34(2):1318–1332, 2008.

[3] I. Garcia, L. Sebastia, E. Onaindia, and C. Guzman. e-Tourism: a tourist recommendation and planning application. *International Journal on Artificial Intelligence Tools (WSPC-IJAIT)*, 18(5):717–738, 2009.

[4] Y. Kurata. Ct-planner2: More flexible and interactive assistance for day tour planning. In *Information and Communication Technologies in Tourism 2011 - Proceedings of the International Conference in Innsbruck, Austria, January 26-28, 2011*, pages 25–37, 2011.

[5] Y. Kurata and T. Hara. Ct-planner4: Toward a more user-friendly interactive day-tour planner. In *Information and Communication Technologies in Tourism 2014 - Proceedings of the International Conference in Dublin, Ireland, January 2014*, pages 73–86, 2014.

[6] G. Laporte and S. Martello. The selective travelling salesman problem. *Discrete Applied Mathematics*, 26(2-3):193–207, 1990.

[7] I. Mínguez, D. Berrueta, and L. Polo. Cruzar: An application of semantic matchmaking to e-tourism. *Cases on Semantic Interoperability for Information Systems Integration: Practices and Applications. Information Science Reference*, 2009.

[8] A. M. Ráez, J. M. Perea-Ortega, M. A. G. Cumbreras, and F. M. Santiago. Otiŭm: A web based planner for tourism and leisure. *Expert Syst. Appl.*, 38(8):10085–10093, 2011.

[9] F. M. Santiago, F. J. Ariza-López, A. M. Ráez, and L. A. U. López. Geoasis: A knowledge-based geo-referenced tourist assistant. *Expert Syst. Appl.*, 39(14):11737–11745, 2012.

[10] S. Staab, H. Werthner, F. Ricci, A. Zipf, U. Gretzel, D. R. Fesenmaier, C. Paris, and C. A. Knoblock. Intelligent systems for tourism. *IEEE Intelligent Systems*, 17(6):53–64, 2002.

[11] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden. The city trip planner: An expert system for tourists. *Expert Syst. Appl.*, 38(6):6540–6546, 2011.

Semantic diversification of touristic recommendations

Joan Borràs
Science & Technology Park for Tourism and Leisure
C/ Joanot Martorell, 15. 43480 Vila-Seca, Catalonia (Spain)
Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA)
Universitat Rovira i Virgili. Av. Països Catalans, 26. 43007 Tarragona (Spain)
joan.borras@pct-turisme.cat

Antonio Moreno
Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA)
Universitat Rovira i Virgili. Av. Països Catalans, 26. 43007 Tarragona (Spain)
antonio.moreno@urv.cat

Aïda Valls
Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA)
Universitat Rovira i Virgili. Av. Països Catalans, 26. 43007 Tarragona, (Spain)
aida.valls@urv.cat

ABSTRACT
Recommender systems aim to suggest lists of items that match accurately the user's preferences. In the last years it has been argued that the *diversity* of the recommendations also plays an important role in the overall satisfaction of the user. Increasing the diversity of the suggestions may be beneficial both for the user (that may discover new, unexpected classes of objects) and for retailers (which may increase the visibility and the sales of the less popular items). This paper provides a brief review of the most popular diversification mechanisms and it introduces a new one based on the semantic clustering of the domain objects. A thorough evaluation of the diversification mechanisms on a Tourism recommender has been performed, reaching the conclusion that the new diversification method achieves very competitive levels of precision and recall, while keeping an acceptable computational cost.

Categories and Subject Descriptors
I.5.3 [**Pattern Recognition**]: Clustering – *Algorithms, Similarity measures*

H.4.2 [**Information Systems Applications**]: Types of Systems – *Decision Support*

General Terms
Algorithms, Performance.

Keywords
Clustering, Decision support, Knowledge personalization and customization, Similarity measures.

1. INTRODUCTION
The unstoppable growths of the Information Society and the Social Web have led to a huge increase in the amount of information available through the Internet on any topic. It has been forecasted [1] that 40 zettabytes (trillion GBs) will be generated only in 2020. This volume of information can easily overwhelm the cognitive capacity of any user looking for specific data.

RecSys'15, September 16-20, 2015, Vienna, Austria.

One of the solutions proposed in the Artificial Intelligence field to this information overload is the use of *Recommender Systems* (RS), which may discover automatically the user's personal preferences and needs and take them into account to analyse huge amounts of data, select the information that is relevant for a particular person and present a personalised list of results.

Precision and *recall* are the metrics more commonly used to measure the accuracy of the recommendations given by a RS. The former indicates the percentage of recommended items which are relevant for the user, whereas the later is the proportion of user-relevant items that have actually been recommended. These measures are indeed important to quantify the degree to which the recommended items match the user's interests. However, it may be argued [2] that other factors also have a strong influence on the overall satisfaction of a user with a RS, being the *diversity* of the recommended items one of them [3]. The intuitive idea is that the recommendation of a set of very similar items may technically be very accurate, since all the items may match quite precisely the user's preferences, but at the same time it may also be counter-productive and unsatisfactory for the user. The recommendation of almost identical items (e.g. books of the same genre by a single author) is boring, unengaging and devoid of *serendipity* (the quality of presenting options that surprise the user and permit him/her to discover new items that may also be interesting, like books of the same genre by other authors, or books by a known author that explore other genres).

The main idea of *topic diversification* is to study how a RS can *balance* the provision of accurate recommendations with the suggestion of items that are different enough to attract the attention of the user and improve his/her experience with the system. The equilibrium between accuracy and diversity is not easy to achieve, as the increase in one of them often leads to the decrease of the other one. If the system does not use diversification mechanisms, the recommended items may be too similar and the system may not be very helpful neither for the user nor for the retailer (that aims to sell all the variety of products, not only those that are most popular and well-known by the majority of users). However, suggesting many items that do not match precisely the user's preferences may also decrease the confidence on the RS and lead to its rejection. Some works actually suggest using two different lists, one with the standard recommendations and another one with related but unexpected items [4].

This works focuses on the study *of diversification mechanims*, understood as algorithms that select a small set of items to recommend to the user from a possibly large set of items that have been previously filtered and ordered by the RS according to the user profile. In this paper the main techniques that have been suggested to diversify a set of recommendations are shown, and some variations and a new method based on clustering are proposed. This novel method has a low time complexity and provides a good level of diversity with an insignificant loss of accuracy. All the diversification techniques commented in the paper have been experimentally tested in a personalised recommender of touristic attractions [5].

The rest of the paper is organized as follows. In the next section we briefly review previous works on the diversification of recommendations. Section 3 explains a new semantic measure of similarity between objects, which is later used to measure the diversity of a set of recommended items. Section 4 presents a list of diversification methods that includes some variations of previous techniques and a new one based on clustering. The balance between accuracy and diversity offered by all these methods has been experimentally tested using *SigTur*, an ontology-based personalised recommender of Tourism activities developed in the Scientific and Technological Park for Tourism and Leisure [5]. The results of these tests are detailed on Section 5. The last section makes some final conclusions and presents potential lines of future work.

2. RELATED WORKS

The techniques that have been proposed in the literature to present a varied list of recommendations may be divided into three main categories. The first group, which is the main focus of this paper, consists on the application of a diversification algorithm on the list of results calculated by a standard RS (which have already been selected according to their similarity with the user's preferences). These algorithms basically change the order of the items in the set of recommendations, ensuring that the first items on the list (the ones that will be finally shown to the user) are both diverse and accurate. The second group integrates the analysis of diversity within the actual ranking procedure of the RS, so that both accuracy and diversity are taken into account at the same time. Finally, the last group includes those techniques that do not focus on individual diversity but on *aggregate* diversity (the level of diversification of suggestions of the RS throughout all users). These methods try to make sure that all the items (even those that are new or unpopular) are actually recommended to some users. Some examples of these three categories are commented in the following paragraphs.

One of the first approaches that studied the diversification of a list of recommended items was [6]. In this work the RS starts by building a ranked list L of recommendable items, taking into account the user's preferences. The first item of this list is added to the final list T of items to be recommended. Then, the system analyzes all the items in L and looks for the item that has more *quality*, which is measured by multiplying the similarity of the item to the user's preferences by the diversity of the item with respect to all the items already stored in T. The item with more quality is added to T. This process is repeated until T contains the number of items that the system intends to recommend to the user (typically the size of T is small –8 or 10 elements- whereas L may have hundreds of items). This algorithm is computationally expensive, since the diversity of each element of L with respect to the set of items already added to T must be checked in each iteration; that's why the authors also propose a *bounded* version of the algorithm, in which only the first items of L are analysed in each iteration. Another work ([3]) added a parameter to this algorithm that permits to adjust the desired level of diversity. In this way the designer of the RS may decide to have more accuracy or more diversity in the offered recommendations, depending on the specific domain of application. In this work each item is represented with a set of attributes, and the values that these attributes can take are structured in a taxonomy. This fact allows the computation of the semantic similarity between pair of items. Another approach in which the level of diversity may be adjusted is reported in [7]. In this work the domain items in L are clustered, taking into account the ratings given by the users. They only consider one element of each cluster in each iteration of the selection procedure; therefore, the computational cost is much lower than the one of the previous methods. Their results show good levels of diversification with a small decrease in accuracy. Another approach of the same family is presented in [8], in which an optimisation method that maximizes the diversity of the recommendation set while keeping an adequate level of accuracy is proposed. The optimisation problem is solved by reducing it to a trust-region problem.

All the works mentioned on the previous paragraph focus on increasing diversity by selecting carefully a set of items from a ranked list of options, previously computed by the RS in some way (usually with a content-based or a collaborative filtering procedure). Other approaches add the diversification mechanisms within the actual ranking procedure of the RS. For instance, Vargas [9] is inspired by diversification techniques used in Information Retrieval, in which results associated to different meanings of the query are shown to the user. His idea is that a set of diverse recommendations may be obtained by showing to the user the results suggested by different recommendation mechanisms. In [10] it is stressed that the selection of an appropriate recommendation technique for a particular user in a specific context is crucial to provide satisfactory results, as the same user may be interested in precise or diverse recommendations in different settings. The same authors propose in another work two similarity measures, *topicality* and *topical diversity*, that may be used to assess the degree of variety of a set of results [11]. They conclude that the aggregation of these similarities offers results with a good trade-off between accuracy and diversity. Zhou introduces a recommendation algorithm called *heat-spreading*, inspired on the physical process of heat diffusion [12]. The idea is to propagate the values of the history of objects evaluated by a user to its neighbourhood. A combination of this method with a classical one focused on accuracy gives results that, in some cases, produce gains both in accuracy and in diversity. Another proposal ([13]) considered the degrees of serendipity and unexpectedness of each item within the recommendation process. The former represents the dissimilarity of the item with respect to the user profile, whereas the later measures the uncommonness of the attribute values of the item within the whole item set. Some authors [14] have pointed out that it is more probable to offer serendipitous results when the RS does not have a large confidence on the information about the user preferences.

The last type of techniques tries to offer *aggregate* diversity, not *individual* diversity. Thus, the aim is to provide a diverse set of recommendations globally, taking into account all the users of the system. These systems are mainly based on collaborative filtering. For instance, Niemann and Wolpers [15] define a notion of similarity between items that takes into account not only their direct co-ocurrence in the purchasing list of users, but also their *second-order* co-ocurrence (two items are similar if each of them appears frequently with a third common item). Therefore, this method finds new links between items that were never bought together. The rating predictions for unfrequent items are increased, hence improving the aggregate diversity. The work reported in [16] proposes different ways of increasing the weight of the items that have been less frequently rated, in order to try to improve their chance of being recommended and increase the aggregate diversity of the RS. One of them is to rank in an ascending order the items based on their number of ratings, from the *lowest* to the highest, so that the most unusual items appear on the top positions. A minimum rating value is set to avoid recommending bad items. Their best results range from a diversity gain of up to 20-25% with only a 0.1% accuracy loss, up to a 60-80% diversity gain with a 1% accuracy loss. Another example of aggregate diversity is proposed in [17]. The main idea is to adjust the similarities between users with a power function to reduce the adverse effects of popular items in user-based collaborative filters. With this method the influence of the most similar users is enhanced, and an increase in both accuracy and diversity is reported.

In this work we want to study the influence of several diversification mechanisms on the results of a personalised recommender of Tourism activities, developed in previous works [5]. Thus, the rest of the article will focus only on the analysis of methods of the first family, which select the items to be shown to the user from the ranked set of options calculated by the RS. Aggregate diversity will not be considered, since the aim is to show to each individual user of the recommender system a varied set of alternatives (keeping a good level of accuracy). However, in the Tourism domain it is also important to make sure that all the activities available on a given area are recommended to some customers (even those that are not very popular), so we intend to include a more detailed study of aggregate diversity and serendipity in our future work.

In section 4 we describe the basic diversification mechanisms proposed in the literature, some variations and a new one based on semantic clustering. Before that, in the next section we describe the semantic similarity measure that will be used to assess the degree of diversity of the items in a list.

3. SEMANTIC SIMILARITY MEASURE

In order to implement a diversification algorithm it is necessary to know how similar (or, actually, dissimilar) two objects are. The use of domain knowledge, in the form of an *ontology*, permits to define *semantic* similarity measures. An ontology is a knowledge structure that represents, in an explicit and formal way, the manner in which a certain domain of interest may be conceptualised. Its main components are *concepts* (classes of objects that share a common property), taxonomic and non-taxonomic *relationships* between them, and *instances* (specific objects of the domain). For instance, Fig. 1 shows a small portion of an ontology of Tourism activities [5]. The concepts shown in the figure are taxonomically related (e.g. *WineRoutes* is a subclass of *GastronomyRoutes*, which is in turn a subset of *Routes*). Each instance (in this case, each particular touristic activity) will be associated to a set of classes; for example, a concrete enological route on a horse could be related to the classes *WineRoutes* and *HorseRiding*. Intuitively, the shorter is the taxonomical distance between two concepts in the ontology, the more similar they are. Following the same example, a touristic route themed on oil (tagged as an *OilRoute*) should be more similar to an enological route (classified as *WineRoute*) than to a tour taken on bycicle (labelled with the *Biking* tag).

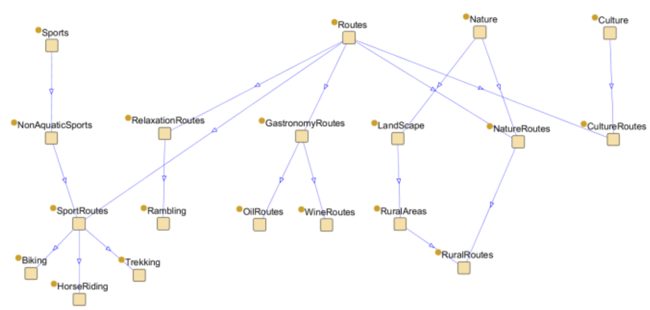


Fig. 1. Portion of a Tourism ontology

This intuitive notion of semantic similarity may be implemented in different ways. One possibility is to count the number of links between two items (e.g. 2 from *OilRoutes* to *WineRoutes*, but 4 from *OilRoutes* to *Biking*). Another possibility, which is the one that will be used in this paper, is to consider the number of shared ancestors between two items (e.g. *OilRoutes* and *WineRoutes* have 2 common ancestors, whereas *OilRoutes* and *Biking* only have 1 common ancestor), as given in [18], [19] and [20]. The *ontology-based semantic distance* (OSD) between two concepts t_i and t_j (1) is measured as the square root of the ratio between the number of different ancestors and the total number of ancestors of both concepts. This distance ranges from 0 (the distance between a concept and itself) to 1 (the distance between two concepts that do not have any common ancestor). In this equation *A(t)* is the set that contains the concept *t* plus all its ancestors (super-classes).

$$OSD(t_i,t_j) = \sqrt{\frac{|A(t_i) \cup A(t_j)| - |A(t_i) \cap A(t_j)|}{|A(t_i) \cup A(t_j)|}} \quad (1)$$

The *ontology-based semantic similarity* (*OSS*) between two concepts is defined as the inverse of the *OSD* (*1-OSD*). The less common ancestors between two concepts, the larger is the distance between them (and the lower is their similarity).

We want to consider the case in which each recommendable item may be associated not only to a single class of the ontology but to a *list* of classes. Thus, we need to define a similarity measure between lists of concepts. Given two lists, the idea will be to measure their resemblance by somehow aggregating the pairwise similarity between the items in both lists. For instance, a simple option could be to take the average similarity between the pairs of concepts. However, this option may return the same aggregated result on very different lists (e.g. the result 0.5 would be obtained with the lists of similarities (0,0,0,1,1,1) and (0.5,0.5,0.5,0.5,0.5,0.5)). In this paper we propose to use the *Ordered Weighted Aggregation* (OWA) family of operators [21] to aggregate the pairwise similarities between the members of two lists. An OWA aggregator is defined with a mapping $R^n \rightarrow R$ that has an associated weighting vector W of dimension n with $\sum_{j=1}^{n} w_j = 1$ and $w_j \in [0,1]$, so that

$$OWA(t_1,...,t_n) = \sum_{i=1}^{n} w_i t_i \quad (2)$$

Thus, the similarity between item a (associated to a set of concepts a_i) and item b (associated to a set of concepts b_j) may be calculated as follows:

$$sim(a,b) = OWA(\{\forall a_i : \max_{\forall b_j} OSS(a_i,b_j)\} \bigcup \{\forall b_j : \max_{\forall a_i} OSS(b_j,a_i)\}) \quad (3)$$

Thus, first we calculate, for each concept associated to item a, which is the most similar concept in b, and this maximum similarity is stored in a list. After that, we repeat the process for all the concepts related to b, and the maximum similarities to concepts in a are added to the same list. Finally, all these values are aggregated, using the OWA operator, into a single final similarity value. The weighting vector regulates the desired degree of andness/orness to be used in the aggregation.

In the diversification algorithms used in the next section it will also be necessary to compute the similarity of an item a with respect to a list l of items (to decide whether the new item is different enough from all the items in the list to be added to it). In this case, we will also apply an OWA operator to aggregate the similarities between the item and each of the members of the list:

$$sim\_list(a,l) = OWA(\forall l_n : sim(a,l_n)) \quad (4)$$

In this expression l_n are the items of the list l and *sim* is the formula used in (3).

4. DIVERSITY METHODS

As shown in section 2, there are several methods that try to improve the diversity of the results offered by a RS. This paper will focus on those methods that, given a long ranked list of alternatives (already ordered according to their relatedness with the user's preferences), decide which (small) set of items will be finally shown to the user. During this selection the system should tend to choose those items that are at the top of the initial list (which are the most accurate), but it should make sure that the selected items are different enough to show a varied set of recommendations.

This section presents the following methods, which in the next section will be evaluated in a Tourism recommender system and discussed in terms of diversity, accuracy and computational cost:

- Baseline-1 [*None*]: just select the top elements of the list, without evaluating their diversity.
- Baseline-2 [*Random*]: select randomly some elements of the list, without evaluating their diversity.
- *Quadratic*: select iteratively the element of the list with the best balance between accuracy and variety with respect to the already chosen items [6].
- *Linear*: variation of the previous method in which a single analysis of the list is made, selecting those items that are different enough from the previously chosen ones.
- *Quadratic break*: variation of the previous method, in which the analysis of the list restarts from the first element each time that an item is selected.
- *Bounded quadratic*: variation of the previous *quadratic* method, in which only the initial elements of the list are taken into account in the selection process.
- New methods based on clustering (*clustering random* and *clustering quadratic*): variations of the *random* and *quadratic* methods in which the elements of the list are clustered (according to their semantic relatedness) before starting the selection process.

The following subsections describe each of these methods, giving an intuitive explanation and the high-level pseudo-code.

4.1 None

This method merely recommends the top N items of the ranked list of alternatives, without evaluating their diversity. Thus, it will serve as a first baseline, as its results will have the maximum accuracy but the minimum diversity.

4.2 Random

This method just selects randomly N items from the initial list, without taking into account neither their position in the list nor their diversity. Thus, both the accuracy and the diversity of the results are unpredictable. This method will be considered as a baseline with respect to which the other diversification methods may be compared.

4.3 Quadratic

This method (see Algorithm 1) tries to find the elements that offer a best balance between accuracy and diversity [3], [6]. In each iteration it loops the whole initial list to find the item that has the maximum combination of accuracy (i.e. the maximum score with respect to the user profile) and diversity with respect to the current *topN* list of selected items. A parameter λ, which ranges between 0 and 1, permits to adjust the desired level of diversity. If it is equal to 0, only accuracy will be considered (i.e. the first N elements of the initial list would be selected, as in the *None* method). If it is equal to 1, it would choose in each iteration the element that is more different from the already chosen ones, regardless of its position in the ranked list.

In line 1 the first item of the ranked list is moved to the *topN* list. This item is the one that has the maximum accuracy. Then the algorithm makes N-1 iterations of a loop. In each iteration the element of L that offers a best tradeoff between accuracy and diversity is selected and added to *topN*. In line 6 the algorithm computes the semantic distance (the inverse of the similarity measure shown in (4)) between each item of L and the whole set of elements already included in *topN*. Then, in line 7 this distance is combined with the weight of the item (i.e. the normalised score given to the item by the RS, which measures how well it fits with the user's preferences) to determine its overall score (which depends on the desired level of diversity). After having analysed all the items in L, the best one is added to topN (line 13) and the method proceeds to the next iteration.

Algorithm 1. Quadratic

Input: *L*: list of items ranked by accuracy, *N*: number of items to recommend, λ: level of diversity

Output: *topN*: list of N items to recommend

```
1:  topN[0] = pop first item from L
2:  n = 1
3:  max = 0
4:  while n < N do
5:     for each item i in L do
6:        d=sim_list(i,topN)
7:        q = ( λ * d ) + ( (1 - λ) * weight of i )
8:        if (q > max) then
9:           max = q
10:          best_item = i
11:       end if
12:    end for
13:    topN[n] = best_item
14:    n = n + 1
15: end while
```

4.4 Linear

This method tries to reduce the computational cost of *Quadratic*, which scans the whole list *L* in each iteration of the selection process. The idea is to make a single scan of the list. When an element that is different enough from those that have already been selected is found, it is added to *topN* and the system continues the analysis of *L* from that point (it does not start again from the beginning, as in the previous method). This behaviour is shown in Algorithm 2.

Algorithm 2. Linear

Input: *L*: list of items ranked by accuracy, *N*: number of items to recommend, λ: level of diversity

Output: *topN*: list of N items to recommend

```
1:  topN[0] = pop first item from L
2:  n = 1
3:  while n < N do
4:     max_distance = 0
5:     for each item i in L do
6:        d= sim_list(i,topN)
7:        if d > λ then
8:           topN[n] = pop item i from L
9:           n = n + 1
10:          if (n = N) then
11:             break for
12:          end if
13:       else if d > max_distance then
14:          max_distance = d
15:          max_item = i
16:       end if
17:       if i is the last item of L then
18:          topN[n] = pop item max_item from L
19:          n = n + 1
20:       end if
21:    end for
22: end while
```

If an element of *L* is distinct enough from the elements already stored in *topN* (condition in line 7), it is immediately added to this list of results (line 8), and the analysis of *L* continues from that point. Notice that in this algorithm the diversity parameter λ is used as a minimum threshold for the distance that an item in *L* needs to have with respect to the items in *topN* in order to be selected. The lower is the desired diversity, the easier it will be for an element of *L* to be selected. The weight of the selected items is not directly taken into account at any moment.

In rare cases, if a very high diversity is required, it might be the case that, after completing a full analysis of *L*, the *topN* list does not contain yet *N* items. If the end of the list *L* is reached, the algorithm adds to *topN* the item that had the maximum diversity with respect to the list of results (line 18) and, if *topN* still does not contain *N* elements, it starts again to analyse *L* from the beginning. This extreme case will not be considered in the posterior study of the computational cost of this algorithm.

4.5 Quadratic Break

The *Linear* method certainly has a much lower computational cost than the *Quadratic* one, since it only makes a single scan of *L*. However, there are cases in which it may present counter-intuitive results. Consider the following example. After adding the first item of *L* to *topN* (line 1 in Algorithm 2), it may be the case that the first element that is different enough from this item is in the 10th position of *L*. After adding this item to *topN*, the algorithm looks (from the 11th position) which is the next item that is different enough from the two items already in *topN*. This item, which is the next one that should be added to *topN*, could be for instance in position 15. However, note that it might be the case that an item in a best position, for instance in position 5, has the same distance to the two items in *topN*. The reason is that, when item 5 was analyzed, it was only compared with the first item in *topN*, because the second item had not been added yet. This example shows that we may select items that have the same (or even worse!) diversity than other items that have a higher accuracy. In order to correct this behaviour, the *Quadratic Break* method goes back to the beginning of *L* every time that it finds an item dissimilar enough from the ones in *topN* (line 10 of Algorithm 3). Thus, the computational cost will be higher than the one of the *Linear* method, although it will not be as computationally expensive as the *Quadratic* one.

Algorithm 3. Quadratic Break

Input: *L*: list of items ranked by accuracy, *N*: number of items to recommend, λ: level of diversity

Output: *topN*: list of N items to recommend

```
1:  topN[0] = pop first item from L
2:  n = 1
3:  while n < N do
4:     max_distance = 0
5:     for each item i in L do
6:        d= sim_list(i,topN)
7:        if d > λ then
8:           topN[n] = pop item i from L
9:           n = n + 1
10:          break for
11:       else if d > max_distance then
12:          max_distance = d
13:          max_item = i
14:       end if
15:       if i is the last item of L then
16:          topN[n] = pop item max_item from L
17:          n = n + 1
18:       end if
19:    end for
20: end while
```

4.6 Bounded Quadratic

The *bounded* version of the *Quadratic* method [6] only takes into account the first $N * B$ items of L (for instance, if the system wants to make N=10 recommendations and B –the *boundedness* factor- is set to 3, the 10 selected items will be taken from the initial 30 elements in L). Intuitively, the results will be more accurate but less diverse, although the computational cost will be heavily reduced because in each iteration only $B*N$ elements will be analysed. The implementation of this method would be exactly like Algorithm 1, except that in the loop in line 5 it would not consider all the elements of L but only those in the first $B*N$ positions.

4.7 Cluster Random

Aytekin and Karakaya proposed the idea of *clustering* the domain items to improve the diversity of the recommendations, by selecting items from different clusters [7]. However, their clustering procedure was based on the *ratings* given by users; thus, it does not assure that the elements of a cluster are semantically similar (very different kinds of items could receive similar ratings). We propose to use this idea, but using a *semantically-based* clustering method. In this way, similar items will be in the same cluster and, if the RS picks up items from different clusters, they will probably be quite diverse.

The clustering of items is made offline using the well-known *k-means* algorithm [22]. The process would be executed periodically to classify new items in clusters. The distance used to group items in each cluster is the *ontology-based semantic distance OSD* defined in section 3. The number of clusters k to be created is application-dependent.

The *Cluster Random* method picks up in each iteration the first element (i.e. the most accurate one) of a randomly selected cluster. The intuitive idea is that the results should be more varied than those of the pure *Random* method, because the elements in different clusters are semantically different. They should also be more accurate, since the selected items are the best ones of their clusters.

The algorithm takes as input the result of the clustering procedure (a list of semantically-related clusters C_1, C_2, C_3, ...). Each cluster contains a list of elements, ordered according to their relatedness to the user's preferences. In each of the N iterations a cluster is randomly selected and its first element is moved to *topN*. The same cluster could be chosen in more than one iteration (note that the number of classes could actually be smaller than N). The aim of this procedure is to select items that have a good accuracy but also offer a good degree of semantic diversity.

4.8 Cluster Quadratic

The idea of the pre-clustering procedure may also be applied to the *Quadratic* algorithm. In this case the computational cost will be heavily reduced, since the iterations are made on the list of clusters rather than on the original list of items, whereas the accuracy and the diversity of the results will be maintained.

The algorithm starts by moving the first item of the ranked list L to the *topN* list (line 1). Thereafter, the algorithm behaves as the *Quadratic* method (Algorithm 1); however, the iterations are made only over the first (i.e. best) items of each cluster. In line 6 the first item of each cluster iteration is considered, and a balanced score of its accuracy and diversity (with respect to the items in *topN*) is calculated (line 8). The element with the best score is selected in each iteration. The computational cost will be much lower than the one of the *Quadratic* method, since the inner loop only considers the k clusters, and not all the elements in L.

Algorithm 4. Cluster Quadratic

Input: L: list of items ordered by accuracy, C: list of clusters C_j (in each cluster items are ranked by accuracy), N: number of items to recommend, λ: level of diversity

Output: *topN*: list of N items to recommend

```
1:  topN[0] = pop first item from L
2:  n = 1
3:  max = 0
4:  while n < N do
5:     for p in 1..k do
6:        i = first item from cluster C_p
7:        d= sim_list(i,topN)
8:        q = ( λ * d ) + ( (1 - λ) * weight of i )
9:        if (q > max) then
10:          max = q
11:          best_item = i; best_cluster=p
12:       end if
13:    end for
14:    topN[n] = pop best_item from C_p
15:    n = n + 1
16: end while
```

4.9 Temporal costs

In Table 1 we show the worst-case temporal cost of each of the methods described in this section. N is the number of items to suggest to a user (in a real case it could be in the 8-10 range). L is the size of the initial ranked list of items calculated by the RS. This size will depend on the database and on the capability of the recommender to filter out the items that do not fit well enough with the user's preferences, but the number of items could be very large (in the thousands). Therefore, L is the parameter that will penalise more heavily the temporal cost. The *Clustering* and *Bounded* methods try to avoid the repetitive analysis of all the elements in L. The cost of the clustering-based methods depends on the number of clusters (C). This number is application-dependent, but it would usually be between 10 and 30. The *Boundedness Factor* (B) should not be a very high number if we desire an efficient bounded method.

Quadratic and *Quadratic Break* are the methods with higher costs, since they depend on $L*N*(N-1)/2$ in the worst case. However, notice that, in a real scenario, *Quadratic Break* will probably have a much lower running time, since it stops every iteration as soon as it finds an element that is different enough from the previously selected ones. The factor $N*(N-1)/2$ appears in several methods because the length of *TopN* increases in each iteration, and we must add the cost of comparing an item with all the elements of *TopN* in each loop. In the *Linear* case it has been assumed that the selected elements are evenly distributed in L. All the methods whose cost depends on L are quite expensive, since L is supposed to be orders of magnitude larger than N, B or C. The *Bounded Quadratic* method reduces the cost with respect to the *Quadratic* one, because $B*N$ should still be much smaller than L. *Cluster Quadratic* is much more efficient than *Quadratic* or *Quadratic Break,* because the number of clusters is much smaller than the number of recommendable items.

It also has to be taken into account that the clustering methods have the additional cost to execute the *k-means* algorithm on the list L to obtain C classes. The complexity of the algorithm can be noted as $O(LCT)$ where T is the (usually small) number of iterations of the process. Since it is a time-consuming process (which, moreover, should be periodically repeated) it should be performed off-line.

Table 1. Table of temporal costs for each diversity method

Method	Cost
None	O (1)
Random	O (N)
Quadratic	O (L*N*(N-1)/2)
Linear	O (L*N/2)
Quadratic Break	O (L*N*(N-1)/2)
Bounded Quadratic	O (B*N*N*(N-1)/2)
Cluster Random	O (N)
Cluster Quadratic	O (C*N*(N-1)/2)

5. RESULTS

This section presents the results of the application of the previous diversification methods on the results offered by *SigTur*, a personalised recommender of cultural and leisure activities [5]. First, we give a brief overview of the recommendation techniques used in this system. After that, a thorough study of the accuracy and the diversity of the results obtained with the approaches described in the last section is reported.

5.1 Personalised recommendation of touristic activities

SigTur is a hybrid RS, which combines a wide range of recommendation techniques (the interested reader is referred to [5] and [23] for more details). *Content-based* methods are used to find out which are the activities that fit better with the user's interests. *Collaborative filtering* techniques provide suggestions based on the items that have been positively rated by similar tourists. *Demographic* data (e.g. country of origin, accommodation type, travel group) are employed to recommend activities that are known to be appropriate for certain kinds of pre-defined users. *Contextual* aspects, like the available time or the budget, are also taken into account in the recommendation process.

Given a certain touristic activity to be evaluated, *SigTur* computes a different score with each of these recommendation methods, which indicates if the item should be recommended to the user or not. These scores are aggregated into a single measure to obtain a final evaluation of each item, used by the system to decide the activities that fit better with the user. A simple average of the scores would not be very appropriate, as they are considering different dimensions of the recommendation problem. *SigTur* uses an aggregation mechanism based on the ELECTRE multi-criteria decision analysis methodology [24]. In this way, it is possible to make sure that only items that have a minimum level of adequacy in the majority of parameters are finally recommended. The aggregated score, which is normalised between 0 and 1, represents the level of accuracy of the recommendation of the item. Thus, *SigTur* calculates a ranked list of the activities that fit better with the user's preferences, demographic data and contextual information.

An interesting aspect of *SigTur* is the dynamic management of the user profile. Initially, the user provides some brief information on the trip and his/her high-level interests. These data, along with basic demographic information, is used to build an initial profile and to provide a first set of recommendations. However, as the user interacts with these initial items the system gathers more information about him/her and it refines dynamically the user profile (e.g. the interest on a certain kind of activities may be raised if the user provides a good rating, asks for more information about an item, or stores a recommended item in the agenda of planned activities). A domain ontology (with more than 300 classes) was specially built for this system (a small portion was shown in Fig. 1). Although there are already some tourism ontologies like [25], our ontology was specially designed to represent as accurately as possible the characteristics of the activities of our database, focusing on a unique *is-a* relationship, rather than other complex relationships. This knowledge structure is used to perform a semantic dynamic update of the user's preferences, as each specific activity is linked to a set of concepts [23].

The diversification methods defined in the previous section have been used to decide, from the ranked list of options computed by *SigTur*, which are the activities that will be finally shown to the user in the screen (a very small subset of the whole set of potentially recommendable activities). Each of the activities is linked to a set of concepts of the ontology. In order to assess the similarity between two activities, or the similarity between one activity and those that have already been selected, the semantic similarity distances defined in equations (3) and (4) were used. Context has not been taken into account in the computation of the similarity (e.g. two History museums are very similar, even if they are located in very different geographical points).

5.2 Evaluation

We want to evaluate how the methods defined on section 4 influence the accuracy and the diversity of the recommendations provided by *SigTur*. It will be considered that the recommendation process employed by the system is correct and it indeed returns a list in which the recommendable items are sorted according to their adequacy to the user. Thus, that initial list is taken to have a 100% accuracy. Each of the diversification methods will choose a subset of the items of the list, decreasing the accuracy but (hopefully) increasing the variety of the results. The final aim is to reach a satisfactory level of diversification with a minimum loss of accuracy (without incurring in a heavy computational cost). The size of the initial list is, on average, 872 elements. Clustering methods group these elements in 23 different clusters. The algorithm selects 8 items to be shown to the user.

The following measures are used to evaluate each of the methods:

- *Diversity*: it is a measure of the pairwise dissimilarity (*1-similarity*) between all the items in the *topN* list (the list of selected items). The similarity between two items is computed with equation (3). The final diversity is computed by applying the OWA aggregator on a vector containing all the pairwise dissimilarities.
- *Precision*: it is computed as the percentage of items in *topN* that are *relevant* for the user. An activity is taken to be *relevant* if it was assigned a minimum score of 0.7 by the ELECTRE method.
- F_{PD}: the F-measure is the harmonic mean of precision and diversity:

$$\frac{2 \times precision \times diversity}{(precision + diversity)} \quad (5)$$

5.3 Comparative analysis of the diversification mechanisms

In the *SigTur* system the factor that has a stronger initial impact on the recommendations is the degree of interest on each travel motivation, explicitly given by the user in a questionnaire. The intuitive idea is that a tourist that sets high values on most of the motivations should be offered a very diverse list of recommendations, whereas a user that only chooses a few motivations is probably interested in visiting more specific places. Therefore, in the experiments shown in this section three different kinds of user profiles have been considered:

1. *General Profile:* the interests on the nine motivations (*Sports*, *Nature*, *Culture*, etc.) are randomly set to values between 70% and 100%.
2. *Medium Profile:* the interests on five randomly selected motivations are set to random values between 70% and 100% (the remaining four motivations are given random interests lower than 30%).
3. *Specific Profile:* two randomly selected motivations are given random interests between 70% and 100%, and the other seven motivations are assigned random interests lower than 30%.

The boundedness parameter for the *Bounded Quadratic* method has been empirically set to 8, after an analysis of the results obtained for different profiles, levels of diversity λ and boundedness factors.

Figs. 2, 3 and 4 show the diversity of the results offered by each of the methods for the three profiles, depending on the desired level of diversity λ. The *Quadratic* and *Cluster Quadratic* methods offer very good results. In the general and medium profiles they start to give diversified results for low values of λ, although for the specific profile they need a higher level of diversity. The *Bounded Quadratic* method offers similar results on the medium profile, but in the general and (especially) in the specific profile it offers lower levels of diversity, even when λ is high. The reason is that the bound cuts off the items at the bottom of the initial list, which are the ones that could offer a high diversity. *Linear* and *Quadratic Break* give similar results, but they require a high level of diversity. Their curve is different from the one of the *Quadratic* and *Cluster Quadratic* methods because the meaning of λ, as described in the algorithms of section 4, is slightly different (in these latter methods it is the weight of diversity with respect to the accuracy, whereas in the *Linear* and *Quadratic Break* techniques is an absolute value of required diversity). The performance of the *Random* and *Cluster Random* methods is not affected by the diversity level, but the diversity of their recommendations varies randomly. The diversity of the results offered by *None* does not depend on λ.

Figs. 5, 6 and 7 show the precision of the recommendations given by the different methods considering general, medium and specific profiles, respectively. The *Random* selection mechanism gives the worst results, as it merely suggests any item of the list. However, clustering the items before the random selection (*Cluster Random*) improves considerably the precision of the results, especially on the general profile. The reason is that items are clustered by similarity, and the best item (i.e. the most accurate) of the selected cluster is retrieved in each iteration. Two methods have a very high precision: *None* (which just returns the most accurate recommendations, without any consideration for diversity) and *Bounded Quadratic*. This method only considers the top *B*N* elements of the initial list to make the selection of the items to be recommended; if most of them have an accuracy over 0.7, the precision will be almost perfect. The remaining methods (*Linear*, *Quadratic*, *Quadratic Break* and *Cluster Quadratic*) reduce their precision when the value of required diversity is increased. If the profile is more specific, the precision decreases more quickly, even from low values of λ. The methods that are more influenced by the diversity level are *Linear* and *Quadratic Break*, whereas *Quadratic* and *Cluster Quadratic* are not so affected by high values of λ.

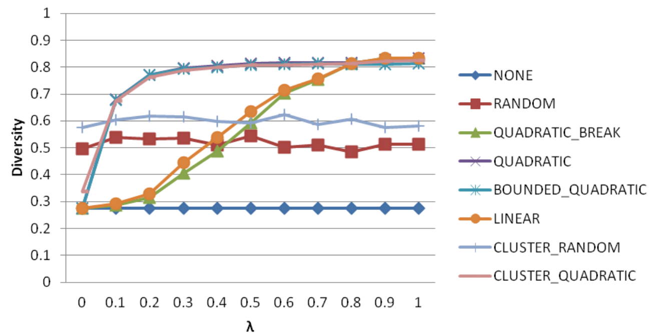


Fig. 2. *Diversity* for the *General Profile*

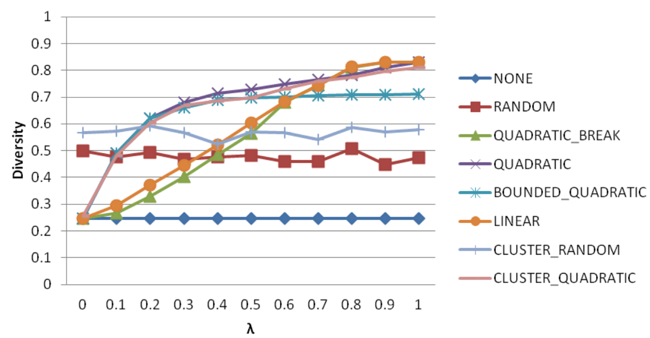


Fig. 3. *Diversity* for the *Medium Profile*

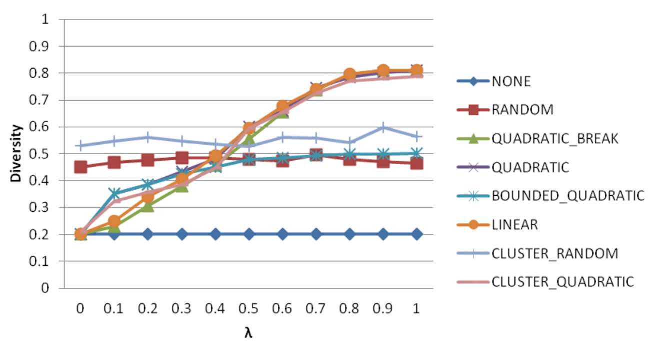


Fig. 4. *Diversity* for the *Specific Profile*

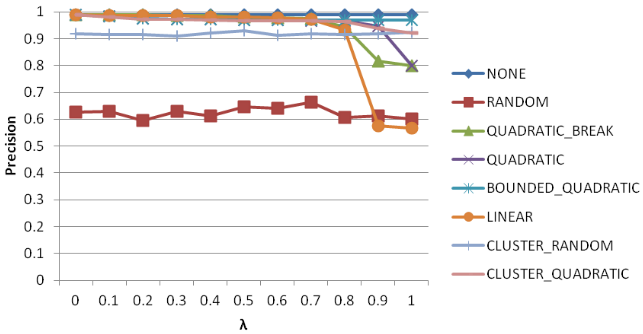


Fig. 5. *Precision* for the *General Profile*

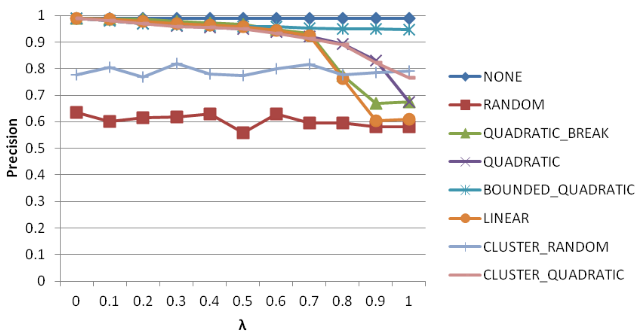


Fig. 6. *Precision* for the *Medium Profile*

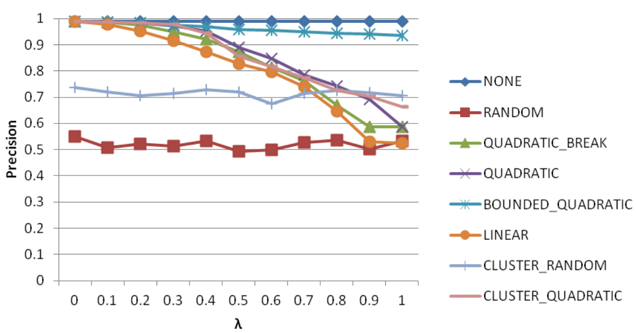


Fig. 7. *Precision* for the *Specific Profile*

The previous figures have confirmed the intuition that, the higher is the value of the required diversity λ, the higher is the diversity and the lower is the precision of all the methods. As the objective is to have high levels in both dimensions, we are interested in analysing the behaviour of the F_{PD} measure, which provides a value that summarizes the global performance of the recommendation method. Figs. 8, 9 and 10 show the results of the methods for the three kinds of profiles. Clearly the *None* and *Random* method offer the worst results. The former has a perfect precision, but its overall performance is heavily penalised by its lack of consideration of the diversity of the results. The latter does not guarantee either accuracy or diversity. As previously commented, a *clusterisation* of the items before the random selection improves the precision (and, therefore, the overall performance) of the method, especially for general profiles.

The method that seems to offer a best combination between precision and diversity across a wide range of required diversity levels is the *Quadratic* one. As seen in section 4.3, this mechanism analyzes all the options in each iteration and selects the one that offers a best compromise between these two perspectives. *Cluster Quadratic* and *Bounded Quadratic* also offer very competitive results, especially in the case of general profiles. This latter method does not decrease its performance for high values of λ, because the bound acts as a roof on the achievable degree of diversity. Finally, the performance of the *Linear* and *Quadratic Break* mechanisms is hampered by their lack of precision, especially in general profiles, because they select the first item that has enough diversity with respect to the previously chosen ones, without taking into consideration its accuracy.

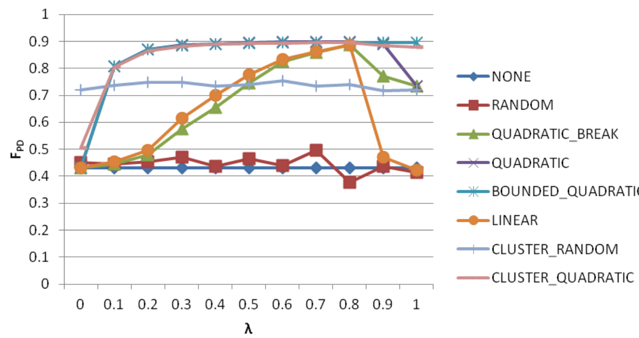


Fig. 8. F_{PD} for the *General Profile*

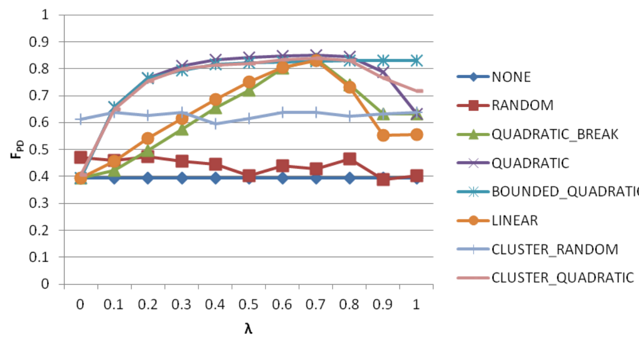


Fig. 9. F_{PD} for the *Medium Profile*

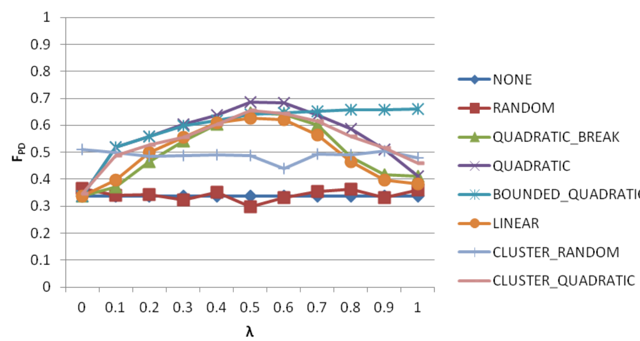


Fig. 10. F_{PD} for the *Specific Profile*

The figures shown above may be used to automatically determine which value should be given to λ to obtain the best results for a particular user, depending on his/her degree of interest in the different travel motivations. It may be seen in the previous figures that the best value for a general profile should be around 0.7, whereas a medium profile gets the best results for values between 0.5 and 0.7 and a specific profile needs a low level of diversity (between 0.2 and 0.3) to offer an acceptable performance. Hence, the degree of diversity that the system should use depends on the kind of user, which can be determined by counting how many motivations the user is interested in. Therefore, the value of λ may be set dynamically with the following formula, where *#chosen_motivations* is the number of motivations in which the user has shown an interest above 30% and *#motivations* is the total number of available motivations (9 in SigTur):

$$\lambda = 0.25 + \left(\frac{\#chosen\_motivations}{\#motivations} \times 0.5 \right) \quad (6)$$

Finally, we show the results of the analysis of 270 user profiles with random motivation values (90 of each kind). The parameter λ is dynamically set for each profile as described in the previous paragraph. Figs. 11, 12 and 13 show, for each diversification mechanism, the averaged results for *Diversity*, *Precision* and F_{PD}, respectively. The diversity in the initial results (without any selection process) is very low (0.24). A simple Random choice already doubles the diversity (0.5). There are 4 methods that offer a level of diversity between 0.57 and 0.65: *Cluster Random, Quadratic Break*, *Linear* and *Bounded Quadratic*. *Quadratic* and *Cluster Quadratic* are the ones that offer highest diversity with values of 0.72 and 0.70 respectively. All the methods offer a precision over 0.9, except *Random* and *Cluster Random*. *Bounded Quadratic* offers better results than *Quadratic*, *Quadratic Break*, *Cluster Quadratic* and *Linear* because the bound puts a limit in the achievable diversification, improving its accuracy. Looking at the global F_{PD} results, the 3 methods that offered more diversity have values around 0.8 (*Quadratic* (0.81), *Cluster Quadratic* (0.79) and *Bounded Quadratic* (0.78)). Two methods slightly exceed 0.7 (*Linear* and *Quadratic Break*), and even *Cluster Random* has a result well above the two baseline methods *None* and *Random*.

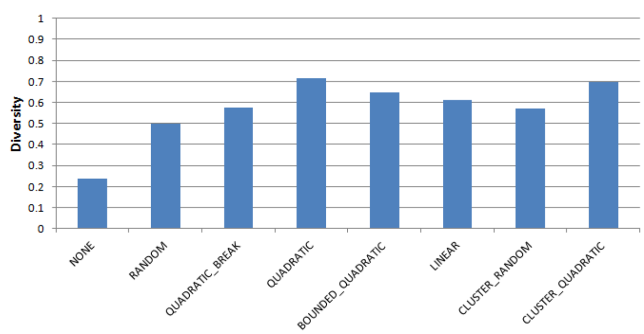


Fig. 11. *Diversity* with dynamic λ

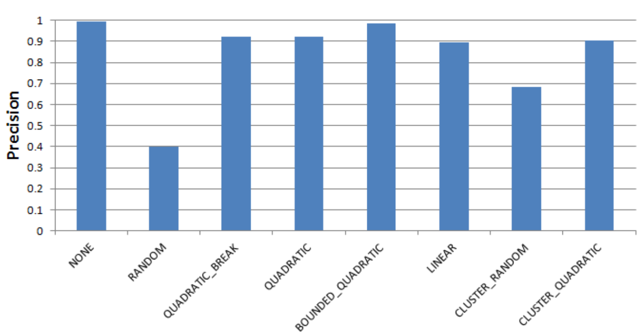


Fig. 12. *Precision* with dynamic λ

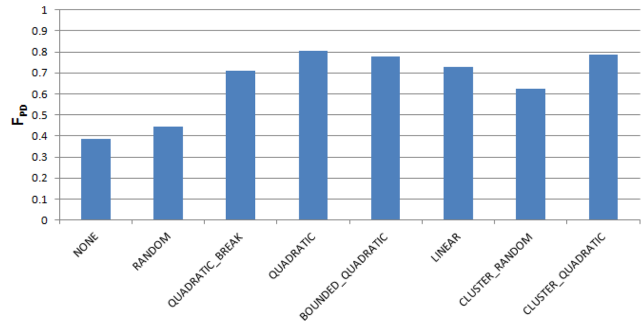


Fig. 13. F_{PD} with dynamic λ

These results must be compared with the time required by each of the algorithms. Fig. 14 shows the number of iterations of each method, except for the *Quadratic* method which, has an extremely high computational cost, giving up to 250,000 iterations and hence exceeding the scale of the chart. The cost of *Linear* and *Quadratic Break* also depends on the size of the initial list, hampering their performance. *Bounded Quadratic*, despite the bound, also has a very high cost. The new method proposed in this paper, *Cluster Quadratic*, seems the best overall alternative, since it provides almost the same performance level and it has a much lower computational cost. Note that these figures do not include the temporal cost of the clustering procedures in *Cluster Random* and *Cluster Quadratic*, which are assumed to be made off-line.

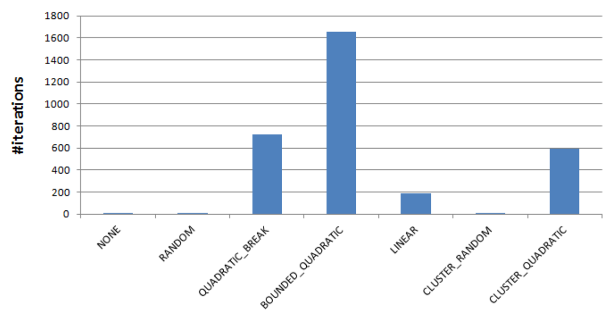


Fig. 14. Number of iterations for each method (except Quadratic)

6. CONCLUSIONS AND FUTURE WORK

In this paper we have described a family of diversification methods, based on the selection of some items from an initial list of recommendations computed by the system. Some variations of previous methods and a new selection algorithm based on semantic clustering have also been proposed. All the methods have been thoroughly tested in an ontology-based personalised recommender of touristic activities [5].

The results of the tests show that the *Quadratic* method is the one that gives the best combination of diversity and accuracy. The main reason is that it loops for all the items of the list to find the item that best combines both diversity and accuracy. However, it is not suitable to be run on real time since its computation costs are extremely high. The remaining methods try to reach similar results more efficiently. For instance, a limitation on the number of items to loop is given on the *Bounded Quadratic* method. Despite the important time reduction with respect to the basic *Quadratic* method, it is still way more expensive than the rest of the methods (see Fig. 12). *Lineal* and *Quadratic Break* try to reduce its computational costs without needing to find the best combination, stopping the selection process whenever they find an item that offers enough diversity. Finally, the novel *Clustering Quadratic* method reduces heavily the computation cost by pre-grouping semantically similar items. Then, the selection loop can be performed through the clusters, and not through the much longer list of items. Moreover, it may be argued that the clustering methodology is more scalable and adaptable to other datasets since the clustering process is based on the semantic similarities between items.

In this paper we have also proposed to dynamically adapt the level of diversification depending on the initial general preferences of the user. Hence, for *generic* users, i.e. those that have a wide range of interests, the degree of diversity can be high since they are willing to accept more diverse items. On the other hand, in the case of those users that are interested on a more concrete set of topics, the degree of diversification should be much lower.

In the future work we want to explore the other two families of diversification mechanisms (see section 2). To evaluate those methods that integrate recommendation and diversity we plan to include diversification mechanisms within the recommendation algorithm of *SigTur*. The idea would be to include in the ranking process of each item some measure of serendipity (or unexpectedness), hoping that the inclusion of serendipitous results will increase the overall satisfaction of the user (a more explicit way to measure this output, either explicitly or implicitly, should also be devised). The study of the methods that offer aggregate diversity is also very interesting from the Tourism point of view, because Destination Management Organisations are very keen on diversifying the tourist offer and increasing the flow of tourists in the less popular and well-known attractions. Finally, a new research line would be to apply the presented diversity algorithms to generate new diverse content, such as for instance, obtain both similar and diverse users in collaborative filtering methodologies.

7. ACKNOWLEDGMENTS

This research has been partially funded by the Spanish research project SHADE (TIN-2012-34369: Semantic and Hierarchical Attributes in Decision Making) and by the SigTur/E-Destination project (FEDER European Regional Funds and the Government of the Province of Tarragona). The work has been supported by the Science and Technology Park for Tourism and Leisure (Vila-Seca).

8. REFERENCES

[1] Gantz J, Reinsel D (2012) The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the Middle East. IDC report.

[2] Mcnee S, Riedl J, Konstan J (2006) Accurate is not always good: How Accuracy Metrics have hurt Recommender Systems, Conference on Human Factors in Computing Systems, Quebec, Canada. pp. 1-5.

[3] Ziegler C, McNee S, Konstan J, Lausen G (2005) Improving recommendation lists through topic diversification. Proceedings of the 14th international conference on World Wide Web, pp. 22-32, ACM.

[4] Ge M, Delgado-Battenfeld C, Jannach D (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In Proceedings of the fourth ACM conference on Recommender systems (pp. 257-260), ACM.

[5] Moreno A, Valls A, Isern D, Marin L & Borràs J (2013). Sigtur/e-destination: ontology-based personalized recommendation of tourism and leisure activities. Engineering Applications of Artificial Intelligence, 26, 1, pp. 633-651.

[6] Smyth B and McClave P (2001) Similarity vs. diversity. In D.W. Aha, & I. Watson (Eds.), Proceedings of the 4th international conference on case-based reasoning. Lecture Notes in Computer Science, 2080, pp. 347–361, Vancouver: Springer.

[7] Aytekin T, Karakaya M (2014) Clustering-based diversity improvement in top-N recommendation. Journal of Intelligent Information Systems, pp. 1-18.

[8] Zhang M, Hurley N (2008) Avoiding monotony: improving the diversity of recommendation lists. Proceedings of the 2008 ACM conference on Recommender systems, pp. 123-130.

[9] Vargas S, Castells P, Vallet D (2011) Intent-oriented diversity in recommender systems. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM, pp. 1211-1212.

[10] Candillier L, Chevalier M, Dudognon D Mothe J (2011). Diversity in Recommender Systems: bridging the gap between users and systems. In CENTRIC 2011, The Fourth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, pp. 48-53.

[11] Candillier L, Chevalier M, Dudognon D, Mothe J (2012). Multiple Similarities for Diversity in Recommender Systems. International Journal On Advances in Intelligent Systems, 5(3 and 4), pp. 234-246.

[12] Zhou T, Kuscsik Z, Liu JG, Medo M, Wakeling JR, Zhang YC (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences, 107, 10, pp. 4511-4515.

[13] Akiyama T, Obara K, Tanizaki M (2010) Proposal and Evaluation of Serendipitous Recommendation Method Using General Unexpectedness. In Proc. Workshop on the Practical Use of Recommender Systems, Algorithms and Technologies, CEUR, 3-10.

[14] Iaquinta L, Gemmis M, Lops P, Semeraro G, Molino P (2010) Can a recommender system induce serendipitous encounters? E-commerce, IN-TECH, Vienna, pp. 229–246.

[15] Niemann K, Wolpers M (2013). A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 955-963). ACM.

[16] Adomavicius G, Kwon Y. (2010) Improving Aggregate Recommendation Diversity Using Rank-Based Techniques. IEEE Transactions on Knowledge and Data Engineering. 24 (5), pp. 896-911.

[17] Gan M, Jiang R (2013) Improving accuracy and diversity of personalized recommendation through power law adjustments of user similarities. Decision Support Systems, 55, 3, pp. 811-821.

[18] Moreno A, Valls A, Mata F, Martínez S, Marín L, Vicient C (2013) A semantic similarity measure for objects described with multi-valued categorical attributes, Artificial Intelligence Research and Development, Frontiers in Artificial Intelligence, IOS Press, 256, pp. 263-272.

[19] Ontañón S, Plaza E (2012) Similarity measures over refinement graphs.Machine learning, 87, 1, pp. 57-92.
[20] González-Calero PA, Díaz-Agudo B, Gómez-Albarrán M (1999) Applying DLs for retrieval in case-based reasoning. In In Procs. of the 1999 Description Logics Workshop (DI'99). Linkopings universitet.
[21] Yager RR (1988) On Ordered Weighted Averaging Aggregation. Operators in Multicriteria Decision making, IEEE Transactions on Systems, Man and Cybernetics. 18, pp. 183-190.
[22] Hartigan JA & Wong MA (1979). Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society, Series C (Applied Statistics) 28, 1, pp. 100-108.
[23] J.Borràs, A.Valls, A.Moreno, D.Isern. (2012) Ontology-based management of uncertain preferences in user profiles. Information Processing and Management of Uncertainty in Knowledge-Based Systems, Part II. Eds: S.Greco, B. Bouchon-Menier, G.Colletti, M.Fedrizzi, B.Matarazzo, R.Yager. Communications in Computer and Information Science (CCIS) 298, pp. 127-136. Springer-Verlag.
[24] Del Vasto-Terrientes, L., Valls, A., Zielniewicz, P., Borràs, J. (2015). A hierarchical multi-criteria sorting approach for recommender systems. Journal of Intelligent Information Systems, 1-34. DOI: 10.1007/s10844-015-0362-7.
[25] Cardoso, J (2005). E-tourism: Creating dynamic packages using semantic web processes. In W3C Workshop on Frameworks for Semantics in Web Services.

The Next Thing- Connecting the museum visit

Antonino Lo Bue — CNR-ICAR Palermo — lobue@pa.icar.cnr.it

Alan J. Wecker — University of Haifa, University of Trento — wecker@gmail.com

Tsvi Kuflik — University of Haifa — tsvikak@is.haifa.ac.il

Alberto Machì — CNR-ICAR Palermo — machi@pa.icar.cnr.it

Oliviero Stock — FBK-Irst Povo — stock@fbk.eu

ABSTRACT
In this paper we present a methodology to provide visitors, in smart regions, additional cultural heritage attractions based on their prior museum visits using: 1) user models and 2) Linked Open Data. Visitor preferences and behavior are tracked via a museum mobile guide and used to create a visitor model. Semantic models and Linked Open Data support the representation of regional assets as Cultural Objects. The visitor model preferences are exploited using a graph similarity approach in order to identify personalized opportunities for visitors by filtering relevant Cultural Objects. The visitor personality characteristics are used to determine number and type of relevant Cultural Objects.

Categories and Subject Descriptors
J.5.m Computer Applications Miscellaneous. H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous

General Terms
Algorithms, Human Factors

Keywords
Personalization, User Models, Linked Open Data, Recommendation System

1. INTRODUCTION
In this paper[1] we present a methodology to provide visitors, in smart regions, additional cultural heritage attractions based on their prior museum visits using: 1) user models and 2) Linked Open Data. We present a blueprint how semantic models and Linked Open Data (LOD) can support the representation of regional assets in order to identify categories of opportunities for visitors based on different personal characteristics determined by previous visits. Having a broad infobase from which to cull possibilities is an arduous task that can benefit from automation. Due to the overwhelming number of possibilities, it is important to personalize the Cultural Heritage (CH) experience. When considering what is required from a smart, personalized system, it becomes clear that the reasoning process of the system has to focus on identifying opportunities for intervention. When and how to intervene and what information to deliver or service to offer. Having a user model, a context model, and a model of the cultural objects are essential for successful support. These can lead to the interaction of museums and other places of cultural heritage to create mega-tourist experience (similar to Verbke and Rekom [6] concept of the "museumpark") which can have a positive market effect for tourism in the region.

We describe our methodology: First we use exhibits in a museum (we use Castle Buonconsiglio in the Trentino Region as examples throughout this paper) and tag them using semantic concepts. Then a mobile museum guide is used to track visitors. Based on this data a user model is developed consisting of characteristics and preferences. We then use a dataset of Cultural Objects using an ontological representation of the domain to cull opportunities. Visitor Preferences are used to filter which Cultural Objects are relevant, and Characteristics are used to determine whether an event or cultural heritage place is desired. Context is used to filter for proximate locations weather conditions, opening times, etc. Again characteristics are used to determine how best to present this information to the visitor

2. BACKGROUND
2.1 Context of Research
The majority of research in the field of Cultural Heritage has concerned itself with the single museum visit (represented as the onsite visit in figure 1). There has been a small amount of additional research looking at the ecosystem of the single visit [2]. This research has looked at planning, the visit itself and post-visit activities (underlined by the parenthesis around the set of asterisks in figure 1). On the other end of scale there have been a few papers on the topic of looking at the lifelong cultural heritage experience [1, 3]. This is represented by the whole line in figure 1. What we aim for in this present research is connecting one point to the next. That is determining how we can get from cultural heritage experience to the next one (represented by the arrow from Summary to Planning) using recommendation system technologies. In the following sections we review two relevant technology areas: User Modeling and personalization in CH, and Linked Open Data and Semantic relatedness.

Figure 1

[1] This is an enhanced version of a paper that appeared in the PeGov workshop at UMap 2015

2.2 User Modeling

According to Ardisonno et al [2], for more than 20 years, cultural heritage has been a favored domain for personalization research and as soon as mobile technology appeared, it was adopted for delivering context-aware cultural heritage information both indoors and outdoors. For personalization, a system needs to have a model of its user. A number of approaches are possible: Overlay, Feature-based, Content based, and Collaborative filtering. In this proposed methodology we basically use an implicit content based approach, where user interests are represented as sets of words occurring in the textual descriptions of items relevant for the user. Visitors have been observed to behave in certain stereotypical movement patterns [14][10]; patterns such as Butterfly, Grasshopper Ant, and Fish [13]. The use of personality types to tailor software is not new. We use the SLOAN Big 5 characterization as it is standard and much research has been done using it [5]. We focus on two traits we believe are connected to the museum experience: Inquisitiveness, which is a measure of curiosity and Orderliness, which measures thoroughness and the need for structure. Introversion and Extroversion could also play a part in group visits, but is not examined in this research. In addition we posit a connection between movement types and the "identity" types proposed by John Falk [4]. Preliminary ideas for the connection of movement patterns to personality types have been proposed [1].

2.3 Linked Open Data

Public agencies collect organize and manage a vast amount of data. Local and European projects aims to deliver data as freely available, reusable and distributed without any restriction, the so called Open Data. As part of these initiatives, tourism and cultural heritage datasets have been published as Open Data. Semantic Web technologies and in particular the Linked (Open) Data paradigm, introduced by Sir Tim Berners-Lee in 2006 [3], are opening new ways for data integration and reuse, creating a method to make data interoperable at a semantic level. Ontologies formally represent knowledge as a set of concepts and their relationships within a domain. RDF and OWL standards enable the formal representation of ontologies as a set of triples (subject, predicate, object). Ontologies are used to express vocabularies of Linked Data triples. On top of RDF and OWL, the SPARQL Query Language is used to query and retrieve information stored as triples thus allowing and facilitating access to the so called Web of Data. DBpedia , which can be seen as the ontological version of Wikipedia, is the core of the Linked Open Data cloud.

In the Natural Language Processing area, semantic relatedness between terms or concepts can be computed using two main approaches: (1) defining a topological graph similarity using ontologies and computing the minimal graph distances between terms, (2) using statistical methods and word co-occurrence in a corpus and calculat-ing the correlation between words. “WikiRelate!" [8], measures correlation among terms using a graph based distance measure on the Wikipedia categories. The system uses the inverse path length measure as a distance metric for terms correlation. Leal et al [9] present an approach for computing the semantic relatedness of terms using the knowledge base of DBpedia, based on an algorithm for finding and weighting a collection of paths connecting concept nodes. The implemented algorithm defines the concept of proximity rather than the inverse path length distance as a measure of relatedness among nodes. Our methodology is based on the inverse path length measure but we apply this to a graph of ontology terms extracted from DBpedia and used as annotation for Open Data resources. Moreover, we also take into account the concept introduced by Moore et al. [10], that evaluates paths calculating the number of outgoing links of each node, in order to improve the precision of the algorithm.

Examples of research initiatives producing Linked Data for the tourism domain are the TourMISLOD project [15], which provides a core source of European tourism statistics using Linked Data, and the OpeNER Linked dataset [16], which provides accommodation data, and other information, such as a short description and location information in Tuscany. Both approaches focus mainly on accommodation modeling and instance buildings, what are missing from the ontology are items to describe Point of Interests (Cultural, Environmental), Places and Events for the tourism domain.

A good example of a more complete model for the tourism domain is the SigTur/E-destination tourism ontology [17]. This ontology presents good coverage of the major tourism concepts, representing 203 concepts in 5 different areas (Events, Nature, Culture, Leisure, Sports, Towns, Routes and ViewPoints). The ontology is exploited by the SigTur recommender system in order to tag the points of interest with the ontology classes.

3. SYSTEM

The mobile guide, at each position of interest (POI), presents a list of relevant media assets. The mobile guide system logs: the POI, which assets are chosen how long they viewed the asset, and in general how long did they stay at the point of interest. We derive two types of information, the first in order to determine general personal characteristics and the second in order to determine specific topic interests. In general we use movement styles, to predict user characteristics (such as personality). We use time viewing presentations in order to determine user topic preferences

The time viewing presentations at a particular exhibit is taken as an indication of user topic preference by using a normalized form over a certain threshold. This determines the most popular exhibits for the particular user. In the following section we show how we use tools to map these exhibits to particular terms in an ontology. We then use this as indication in our user model of the user's interests.

In order to characterize the user we make use of his general movement activities. We use the following statistics: 1) NumberOfPOIsVisted (NPV) – number of positions where a person stayed more than 9 seconds as detected and logged by the mobile guide's positioning system. Nine seconds is a number we have used for previous analysis and has provided good results. 2) POIsWherePresentationsSeen (PPS) – the number of positions where the visitor viewed at least one media asset connected to that position as computed from the logs of the mobile guide. 3) NumberOfPresentationSeen (NPS) – the total number of media assets the visitor viewed as computed from the logs of the mobile guide. We can think of the ratio of PPS/NPV as measuring the user's curiosity (typical of the Inquisitive personality types) and the NPS/PPS ratio as measuring the user's attention span (typical of the Orderly personality type). Table 1 shows our mapping of movement based on the above formulas to: 1) the types of Veron and Levasseur, 2) SLOAN personality types and 3) Falk identity types.

Table 1. Connecting the user behavior to personality and Falk types

Behavior	*Personality*	*Falk*	*Formula*
Fish	Non curious – Unorderly	Recharger	((PPS/NPV < = T_1) &(NPS/PPS < T_3))
Ant	Inquisitive – Orderly	Explorer	(PPS/NPV > T_1) & (NPS/PPS > T_2)
Grasshopper	Non curious – Orderly	Professional	(PPS/NPV > T_1) & (NPS/PPS < T_2)
Butterfly	Inquisitive – Unorderly	Exp. Seeker	(PPS/NPV < T_1) & (NPS/PPS > T_3)

3.1 Derivations and Matchings

The system uses annotated internal and external information about cultural places and events. Internal information is taken from catalogues or websites and is used by the mobile guide app to describe user preferences by storing the relevant topics related to exhibits the user has visited and liked. External information is imported from available Open Data about museums and cultural events and enriched in the domain ontology, using knowledge from the Linked Open Data cloud (DBpedia dataset). Data is stored using a domain ontology for tourism called *eTourism*[2]. *eTourism*, is a cultural domain ontology that describes services (e.g. Hotels, B&B), points of interest (e.g. Museums, Archeological parks, Libraries) and events. The ontology covers methodological and practical aspect of services (hotels, B&B, etc.), cultural objects (museum, cultural places, etc.) and events. It is used as a vocabulary model to map external Open Data into RDF triples validated by the ontology concepts. For the present work we have developed a specific module of the *eTourism* ontology named *Cultural Objects Ontology* (*coo*) that covers (1) properties (such as topic, keywords, geographical information) of museums or events, exploits the semantic identity with LOD/DBpedia concepts (using *owl:sameAs* predicates) and implements (2) user profile types and topics of interests selections.

For each museum source, we extract - as a first step, keywords from the exhibits (the example we will use is that of the Castle Buonconsiglio museum in Trento). We exploit the semantic relatedness implementing a graph similarity approach. We annotate keywords - for each description, and we disambiguate them to DBpedia concepts using DBpedia Spotlight APIs[3]. We filter out all the non-relevant concepts and then obtain a bag of concepts (related to cultural heritage) similar to the following:

{dbpedia[4]:Trentino, dbpedia:Prehistory, dbpedia:Ancient_Rome, dbpedia:Middle_Ages, dbpedia:Hunter-gatherer, dbpedia:Upper_Paleolithic, dbpedia:Bronze_Age}

In DBpedia, each concept is related to a category using the property *dcterms:subject*, then each category is part of a hierarchy structure with nodes connected via *skos:broader* properties. For example the below two DBpedia concepts have as *dcterms:subject* the DBpedia *topic* categories:

1) Last_glacial_period *(dcterms:subject) ->{Climate_history, Glaciology, Holocene, Ice_ages}*

2) Ancient_Rome *(dcterms:subject) ->{Ancient_history, Ancient_Rome, Civilizations}*

For the second step, we extract from the DBPedia SPARQL endpoint, for each concept, the *topic* categories of the DBpedia taxonomy. As result we obtain a wider bag of DBpedia *topic* categories describing each museum exhibit. Using the hierarchical structure of categories is thus possible to discover similarities among concepts that have ancestor categories in common.

As external sources, we take the Open Data set delivered by the Italian Cultural Heritage Minister[5] (MIBAC) and we map these objects using the *coo* ontology; then, for each object, we exploit the same process applied for the internal resources, in order to annotate and extract the corresponding bag of topics. As a result, we obtain a list of information for each MIBAC *Cultural Object* (cultural place or event), as in the following example:

foaf:name = "Memorie della Grande Guerra",

coo:mainCategory = http://dbpedia.org/resource/Category:History

Bag of Concepts (*dcterms:description*) ->
{1918_disestablishments, Aftermath_of_World_War_I, Austria-Hungary, Austria_articles needing attention, States_and_territories_established_in_1867, Anoxic_waters, Back-arc_basins, Contemporary Italian_history, History_of_Austria-Hungary, History_of_modern_Serbia, Wars_involving_Italy, World_War_I }

In order to select suitable *Cultural Objects* candidates for the user, we define a metric to measure the semantic distance between the user profile tags and the available cultural objects tags. As a first step, we measure the *shortest path distance* between each of the *m topic* categories in the bag of topics of the user profile and the *coo:mainCategory* topic of the suitable candidates (see table 2), and we reduce candidates cardinality by applying an upper threshold to the distance.

Table 2. Example path between two DBpedia categories

Distance	*Steps*
0	dboc:[6]Ancient_history
1	dboc:Periods_and_stages_in_archaeology
2	dboc:Archaeology
3	dboc:Conservation_and_restoration
4	dboc:Art_history
5	dboc:Visual_arts
6	dboc:Arts

[2] Currently under development at ICAR-CNR within the framework of the national project Dicet-InMoto-Orchestra, (http://www.progettoinmoto.it). Ontology documentation is available at http://slab.icar.cnr.it/eTourismLite/

[3] http://spotlight.dbpedia.org

[4] Prefix for http://dbpedia.org/resource/

[5] http://dbunico20.beniculturali.it/DBUnicoManagerWeb/#home

[6] Prefix for http://dbpedia.org/resource/Category:

After this step, we refine the result by calculating (via SPARQL queries on the DBpedia endpoint) the *shortest path* between the user bag of topics (*m)* and the suitable candidates bag of topics (*n)* on the remaining subset of cultural objects.

The following code shows an example SPARQL query used to calculate shortest path between the DBpedia category nodes. The query makes use of Virtuoso Open Source special features in order to manage the step maximum length and transitivity for the *skos:broader* relation edges of the graph. A similar result can be obtained using SPARQL property paths to express a sequence of multiple occurrences of *skos:broader*:

```
#Calculates shortest path in one direction

SELECT ?dist  ?steps
 WHERE {
   ?in skos:broader ?out
OPTION(TRANSITIVE,t_max(15),T_DISTINCT,
T_DIRECTION 2, T_SHORTEST_ONLY, t_in(?in),
t_out(?out), t_step ('step_no') as ?dist, t_step
(?in) as ?steps) .

  FILTER (?in =
<http://dbpedia.org/resource/Category:Geography>)
  FILTER (?out =
<http://dbpedia.org/resource/Category:History>)
  }
ORDER BY ?dist
```

Its important to underline that when computing the distance measure between topic categories we also take into account, for each hop of the shortest path, the number of outgoing links of the node: the more outgoing links a node has (to other DBpedia taxonomy nodes) the less it is specific. Broad connected nodes receive low weights while nodes with less outgoing connection will get higher values. The following code shows an example SPARQL query used to calculate the outgoing connection for a particular node of the graph:

```
#Extract branches skos:broader in and out for a
specific resource
SELECT  (count(?out) as ?out_branches)
  WHERE
  {
   ?res  skos:broader ?out .
   FILTER( ?res =
<http://dbpedia.org/resource/Category:Industry_mus
eums>) .
  }
```

Using each pairwise distance as a component of a normalized vector of distances, we evaluate, for each museum or event an average normalized distance for each *m* user category and we sum all these distances to define the relatedness of each cultural object. Again an empirical threshold on distance is applied to retain a limited number of candidates.

3.2 Use of personality types

Using behavior types or personality types we can tailor the amount and presentation of information. For example for ants and butterflies we can give ten items. For grasshoppers and fish we may only give two items. Ants and grasshoppers may be given places while butterflies and fish may be given events. Additional personalization may be possible.

4. DISCUSSION AND CONCLUSIONS

The results we get for the four sample users are shown on the table below.

Table 3. Simulated output of the system with Places and Events suggested per each user behavior. Suggested items are marked with a *.

Type	*Preferences*	*Places, Events*
Ant	Bronze_Age (.5), Feudalism (.2), Middle Ages (.5), Ancient Egyptian funerary practices (.1), Civilizations (.2)	***Places:*** Museo archeologico dell'Alto Adige (Archeology) (.6), Area archeologica Palazzo Lodron (Archeology) (.6), Museo delle palafitte del Lago di Ledro (History) (.4), Museo locale di Aldino (Etnography) (.2*)
Grass-hopper	Romantic_art (.4), 20th-century Italian_painters (.3), Postmodern_art (.3), Fresco_painting (.3),Rural_culture (.1)	***Places:*** Museo Rudolf Stolz (Arts) (.6), Museo di arte moderna e contemporanea di Trento Rovereto (Arts) (.5), Museion - Museo d'arte moderna e contemporanea (Arts) (.6), Museo della Val Venosta (Anthropology) (.2*)
Butterfly	World_War_I (.4), Civilizations (.4), 1st-century Roman emperors (.2), History_of_Europe (.6), Rural_culture (.2)	***Events:*** Doni Preziosi, Immagini e Oggetti dalle Collezioni Museali (Exhibition/History) (.5), Storie da Trento all'Europa. Mostra documentaria (Exhibition/History) (.5)
Fish	Romantic_art (.4), 20th-century Italian painters (.3), Bronze_Age (.5), Fresco painting (.3), Rural_culture (.1)	***Events:*** Rinascimenti Eccentrici al Castello del Buonconsiglio (Exhibition/Arts) (.7), Apertura Spazio archeologico Sotterraneo del Sas (Opening/Archeology) (.4)

Our current metric of semantic relatedness doesn't take into account whether the user profile bag of topics is representative of a sufficiently broad range of museums categories to cover their cultural preferences. To balance this, when all/most of the user preferences are of the same topic area (e.g. Prehistory), one or more among suggested items could be chosen from a minor topic category, to elicit variation in user interests.

Our current research involves the implementation of the methodology to the Old City and the Tower of David Museum in Jerusalem, and the evaluation of the user model and the semantic suggestions results.

5. ACKNOWLEDGMENTS

Our thanks to the anonymous reviewers for their comments.

6. REFERENCES

[1] Antoniou, A., Lepouras, G., Lykourentzou, I. et al.: Connecting Physical Space, Human Personalities and Social Networks.

[2] Ardissono, L., Kuflik, T., Petrelli, D.: Personalization in Cultural Heritage: The Road Travelled and the One Ahead. User Modeling and User-Adapted Interaction, 22 (2011) 1-27

[3] Berners-Lee, T. (2006). Design issues: Linked data.

[4] Falk, J. H.: Identity and the museum visitor experience. Left Coast Press Walnut Creek, CA (2009)

[5] Higgins, D. M., Peterson, J. B., Pihl, R. O. et al.: Prefrontal Cognitive Ability, Intelligence, Big Five Personality, and the Prediction of Advanced Academic and Workplace Performance. J. Pers. Soc. Psychol., 93 (2007) 298

[6] Jansen-Verbeke, M., & Van Rekom, J.: Scanning Museum Visitors: Urban Tourism Marketing. Ann. Tourism Res., 23 (1996) 364-375

[7] Kuflik, T., Kay, J., Kummerfeld, B.: Lifelong Personalized Museum Experiences. Proc.Pervasive User Modeling and Personalization (PUMP'10), (2010)

[8] Kuflik, T., Wecker, A. J., Lanir, J. et al.: An Integrative Framework for Extending the Boundaries of the Museum Visit Experience: Linking the Pre, during and Post Visit Phases. Information Technology & Tourism, (2014) 1-31

[9] Leal, J. P., Rodrigues, V., & Queirós, R.: Computing semantic relatedness using DBpedia. SLATE 2012, page 3519, 2012.

[10] Moore, J. L., Steinke, F., & Tresp, V.: A novel metric for information retrieval in semantic networks. ESWC 2011 Workshops 65-79

[11] Stash, N., Veldpaus, L., De Bra, P. et al.: Creating Personalized City Tours using the CHIP Prototype. (2013)

[12] Strube, M., & Ponzetto, S. P.: WikiRelate! Computing semantic relatedness using Wikipedia. AAAI Vol. 6, pp. 1419-1424 (2006).

[13] Veron, E., & Levasseur, M.: Ethnographie de l'exposition. Centre Georges Pompidou (1983)

[14] Zancanaro, M., Kuflik, T., Boger, Z. et al.: Analyzing Museum Visitors' Behavior Patterns. User Modeling 2007, (2007) 238-246

[15] Sabou, M., Arsal, I., & Braşoveanu, A. M. (2013). TourMISLOD: A Tourism Linked Data set. *Semantic Web*, *4*(3), 271-276.

[16] Bacciu, C., et al. Accommodations in Tuscany as Linked Data. *LREC 2014*: 3542-3545.

[17] Moreno, A., Valls, A., Isern, D., Marin, L., & Borràs, J. (2013). Sigtur/e-destination: ontology-based personalized recommendation of tourism and leisure activities. *Engineering Applications of Artificial Intelligence*, *26*(1), 633-651.

Design and Development of a Real Time RecSys Based on Location, Mobile Device and Tourists Activities

Seyed AmirMohsen Jazdarreh
University of Greenwich
London, UK
S.jazdarreh@greenwich.ac.uk

Ed de Quincey
Keele University
Keele, UK
E.de.quincey@keele.ac.uk

Lachlan Mackinnon
University of Greenwich
London, UK
L.Mackinnon@greenwich.ac.uk

ABSTRACT

In this demo paper, we describe a novel model for real time recommender systems within popular UK tourist attractions using Canterbury Cathedral as a case study. A mobile application has been developed for tourists that supports access to filtered information of places within the attraction such as artefacts, custom guide, maps, shop information and provides facilities to share visitors' experiences. In addition a real time recommender system based on a tracking mechanism and users' activities with the app has been created to enhance the tourist experience. In this case, visitors will be able to gain relevant contents through physical engagement with artefacts via smart posters and their mobile devices. This combination of technology and user interaction aims to deliver a dynamic, online location-based service for tourists and provide enhanced usage data for the Cathedral managers.

Categories and Subject Descriptors

H.3.5 [**Online Information Service**]: Information Storage and Retrieval.

Keywords

Recommendation system, real time data analysis, tourism, mobile application.

1. INTRODUCTION

Currently, one of the main challenges for the administrators of tourism sites that are visited by large numbers of tourists is providing relevant and easily accessible information about monuments, artefacts, places etc. Tourism therefore has been described as a hybrid industry [1]. Although focused upon the provision of information, it is essentially concerned with physical environments. The physical and digital worlds however are becoming inextricable due to the visitors' requirements to be connected anytime and anywhere [1]. Tourists are using smart devices and apps before, during and after a trip, making this area increasingly more significant from a tourism management perspective [2]. Most mobile apps related to the tourism industry such as Trip-advisor or Trip-IT support users in order to make travel easy and recommend them useful information such as relevant attractions and activities, the lowest ticket prices, and popular hotels and restaurants. These apps are useful in helping users decide where to go or how to get there, but often the interiors of attractions and historical places are neglected in current RecSys apps. Currently, visitors decide themselves what to do when inside attractions as there is no reliable recommender system to assist them, such as a sophisticated tour guide [3].

In this demo paper, we briefly present a mechanism that on one hand increases visitors' physical and virtual engagement in the Cathedral and on the other, collects tourists' digital foot prints which have been left through visitors' interactions with the system. The resulting data is massive and multidimensional (e.g. time, location, area of interests etc.). However this data can be analysed to determine the popularity of certain artefacts and items, enabling attraction managers to make informed decisions about future promotions, and provide different self-development RecSys based on tracking visitors' activities to enhance tourism experiences.

http://dx.doi.org/10.1145/2645710.2645718

2. OVERVIEW

In this section we describe the main structure of the system.

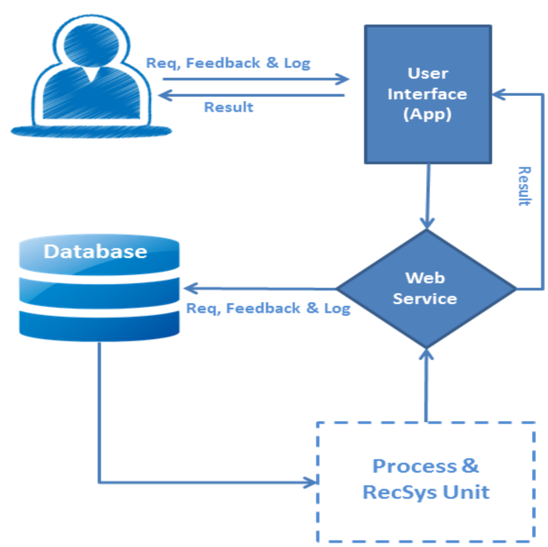


Figure 1. System Architecture.

Visitors interact with the mobile app via smart posters which have been placed next to each artefact. A smart poster is nothing more than a traditional poster which is embedded with Near Field Communication (NFC) that is a short-range wireless technology for data transfer and information exchange. These either visible or hidden NFC tags are programmable and able to keep small amount of data. A web service has been developed to provide a bridge between the NFC smart posters and a database. As shown in Figure 1, when a visitor taps his/her NFC enabled mobile phone on a NFC smart poster, a request is sent to the database via the web service to access the corresponding data about the artefact that smart poster relates to. The returned contents consist of multimedia-enriched data (text, image, map, audio guide etc.)(Figure 2). Besides returned results, a tracking system has been developed to run while visitors interact with the app as a background process. This mechanism is able to record and store users' activities in the database through their journey within the Cathedral. These factual data include:

- Number of tapped smart-posters per each artefact.
- Rating artefact.
- Time capturing.
- App usage (e.g. user's interaction with different features of the app).

User's privacy is not affected as no personal information (no registration) neither required nor used to operate the app. The innovative feature of this RecSys tourism app is the provision of new levels of context awareness and on-demand delivery of information and services to the tourists, by combination of visitors' participation, real time data analysis and the deployment of smart posters.

Later, by monitoring users' logged data (factual data), the site managers can make more informed decisions to provide better services to visitors. We believe this project can open up a number of new opportunities within the Cathedral and other tourist sites.

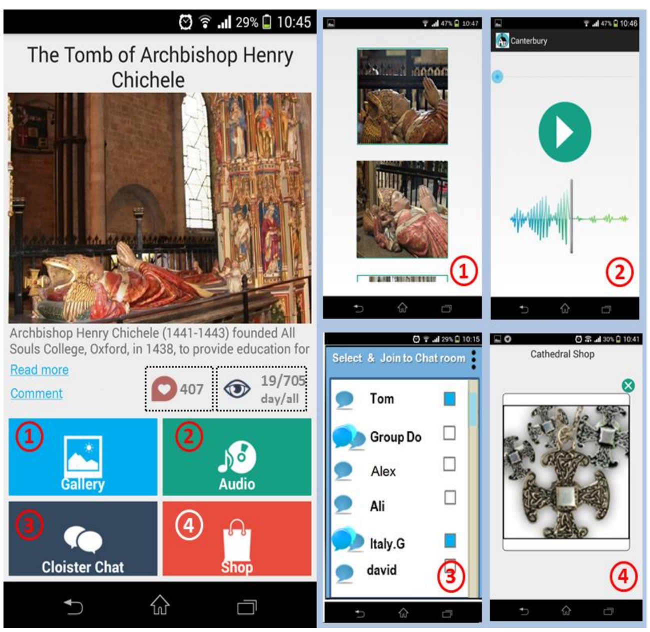


Figure 2. The Mobile Application features

2.1 Process and RecSys

The point of strength in this RecSys is that visitors automatically determine the popularity of artefacts and their contents, simply by tapping on NFC smart posters and leaving their foot prints. Moreover, visitors are able to recommend and communicate with each other using the app's features such as comments and rating. (Figure 2). These recommendations include:

- **The most popular artefacts;** according to the number of visits per item, our system is able to identify the "hottest" objects (per day or totally) and recommend them to visitors.
- **Comments, Rates and Like;** Similar to existing social networking services, users are able to make comments, ratings and leave feedback in the app.
- **Most viewed item (Audio, Image, text) per artefact;** The RecSys will be recommending the most popular content that has been viewed by visitors on a particular artefact.

In addition, visitors who are in the Cathedral are then able to encourage each other to share their live experiences through the chat facility in the same place at the same time. They can select a private or a group chat and start to communicate or make a plan for a discussion or ask questions.

3. CONCLUSION

In this demo paper, we propose a system which can analyse users' interactions with both physical artefacts (via smart posters) and a mobile application. We provide a platform that although supports access to filtered information within the attraction, users do not need to supply personal information and will capture user data e.g. activity implicitly, and will recommend relevant content using this factual data. This model is currently being trialled and evaluated in a popular tourist attraction in the UK, Canterbury Cathedral and further RecSys methods investigated for inclusion.

4. ACKNOWLEDGMENT

This project is supported by an EPSRC iCase funded PhD looking into the potential uses of contactless technology, real-time data analysis and social media capability to support tourism at places such as Canterbury Cathedral. We would like to thank all participants in this project.

5. REFERENCES

[1] Pease, Wayne and Rowe, Michelle (2005) *an overview of information technology in the tourism industry*. In: ITS Africa-Asia-Australasia Regional Conference - ICT Networks - Building Blocks for Economic Development, 28-30 Aug 2005, Perth, Western Australia.

[2] J. Pesonen and E. Horster, "*Near field communication technology in tourism*," Tourism Management Perspectives, vol. 4, pp. 11–18, Oct. 2012.

[3] Sarkaleh, M. K., Mahdavi, M., & Baniardalan, M. (2012). *Designing a Tourism Recommender System Based on Location, Mobile Device and User Features in Museum.* International Journal of Managing Information Technology, 4(2), 13.

Visit Costa Daurada & Terres de l'Ebre: A semantic recommender system of tourist activities

Joan Borràs
Science & Technology Park for Tourism and Leisure
C/ Joanot Martorell, 15. 43480 Vila-Seca, Catalonia (Spain)
Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA)
Universitat Rovira i Virgili. Av. Països Catalans, 26. 43007 Tarragona (Spain)
joan.borras@pct-turisme.cat

Antonio Moreno
Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA)
Universitat Rovira i Virgili. Av. Països Catalans, 26. 43007 Tarragona (Spain)
antonio.moreno@urv.cat

Aïda Valls
Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA)
Universitat Rovira i Virgili. Av. Països Catalans, 26. 43007 Tarragona, (Spain)
aida.valls@urv.cat

ABSTRACT
This paper presents a recommender system designed to provide personalised recommendations of touristic activities in Costa Daurada and Terres de l'Ebre region (henceforth "*Visit CD & TTE*"), in Tarragona, Spain. The activities are properly classified and labelled according to a specific ontology, which guides the reasoning process. The recommender takes into account many different kinds of data: demographic information, travel motivations, the actions of the user on the system, the ratings provided by the user, the opinions of users with similar demographic characteristics or similar tastes, etc. This system can have a beneficial impact on the region by improving the experience of its visitors.

Categories and Subject Descriptors
H.4.2 [**Information Systems Applications**]: Types of Systems – *Decision Support*

General Terms
Algorithms, Design.

Keywords
Semantic Recommender Systems, Ontologies, Decision support, Knowledge personalization and customization, Similarity measures.

1. SYSTEM DESCRIPTION
Visit CD & TTE is a recommender system for tourists that want to plan a visit to the province of Tarragona. This system provides personalised suggestions combining content-based and collaborative filtering techniques. It also applies Artificial Intelligence tools such as automatic clustering algorithms, *Multiple Criteria Decision Aid* (MCDA) methods [1], ontology management [2], semantic diversification and the definition of new similarity measures between users based on complex aggregation operators [3].

The system offers a Web-based interface that facilitates the user interaction and provides a better experience in the travel preparation stage of the trip. Concerning the information used by the recommender, it takes into account demographic data, the travel context (e.g. transport means), geographical aspects, information provided explicitly by the user (*e.g.* main travel motivations) and implicit feedback deduced from the interaction of the user with the system.

After accessing the web site the user is firstly asked to fill up the user profile (Figure 1) which includes the traveller group type (family, couple, friends, alone or business), average age of the group, trip dates, transportation means and 7 motivations (beach, leisure and entertainment, nature, culture, health and care, sports and enotourism). In addition, the user profile is enhanced with the language of the user which is taken to be the language chosen in the web site.

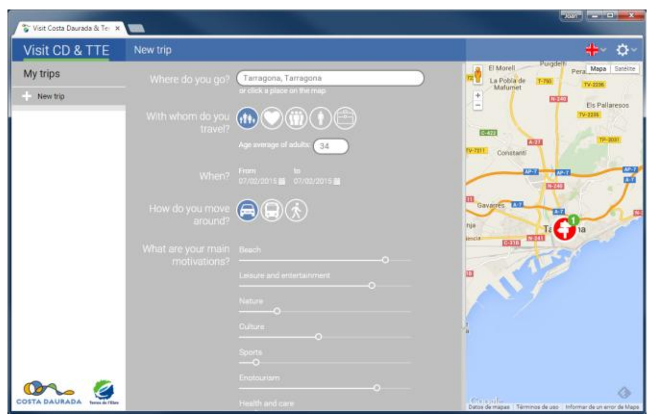


Figure 1. Form to build the user profile with travel motivations and characteristics

Once the user has completed his/her profile, the system uses the recommendation engine to suggest a personalised and diversified list of items [3]. The recommendation process combines content-based methods (managing user preferences using ontologies [2], with the information about the main motivations and the history of interacted items) and collaborative filters providing suggestions given by similar users (the similarity between users can be based on their interaction with the items or on their demographic information). The user also specifies his/her destination, so the system can suggest attractions nearby taking into account the transportation means used by the travellers (walking, car driving and public transport). The times and routes on the map can be customized for any kind of transportation (even public transport can be managed with the Google directions API).

Figure 2 shows the page where the user receives the list of suggestions, which keeps continuously adding new items as the user scrolls down. At the right hand side the map shows items that are on the list, and on the left there is a menu that allows the user to switch between his/her trip (at the top) or to focus on a particular type of activities.

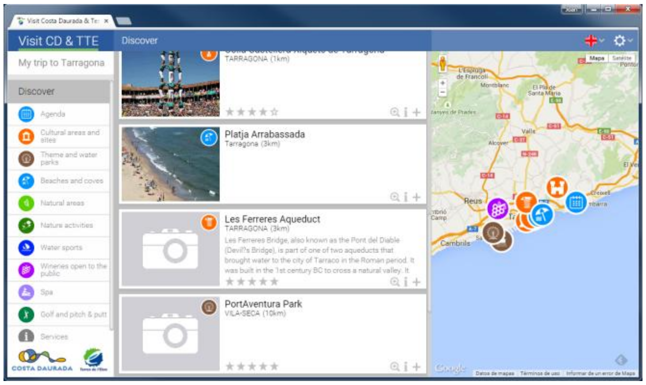


Figure 2. List and map of suggested activities

The system monitors continuously the navigation of the user through the map. When the selected geographic area does not contain any item from the current list, the system adds to the map the items better ranked within such region. The system tends to avoid overcrowding the list and the map with a large number of items, but if the user zooms in the map to a particular area to see one item, the system will push new items on the map. If it is possible, the system will always show at least 6 items on the map, taking into account the ranked list of items within the map region.

The user can select an item to see more information about it or to add it to the travel plan. The planned route is shown when the user clicks on the link of the name of the trip (top-left side of Figure 2). Then the list of suggestions switches with the list of the planned items, as shown in Figure 3. The order of the items to be visited can be arranged manually (drag and drop action) or automatically (with the most efficient path). If an automatic route is requested, the system distributes the activities in the available days depending on their location and their visiting time. Items are printed with numbers on the list and on the map so that the user may follow the planned route easily. The weather forecast application OpenWeatherMap API has been used to print the weather prediction for each day of the trip, thus helping the user to decide if an item should be scheduled or not on a particular day.

Another option available to the users permits them to get inspiration from trips created by other travellers. Therefore, we have included a new social aspect to the system, where users can explore trips of other users and copy the one that fits better with his/her preferences. Since the number of trips created by other users may be high in the future, the system will show a ranked list of trips given by its similarity in terms of demographic attributes (e.g. kind of group) and the characteristics of the trip (e.g. transport means or the number of days of the trip). Figure 4 shows the page in which the user can navigate through similar trips. At the top of the centre panel there is a ranked list with similar trips. In this example we can see that the first trips are located in Tarragona, contain car driving directions and have a length of two or three days, like the trip created by the current user. Language, age and travel group are also taken into account to rank the trips but they are not as restrictive as the other parameters. Below the list of trips there are the details of the route of the selected trip, which the user may explore. These routes can be fully copied to the current trip and thereafter be modified as necessary.

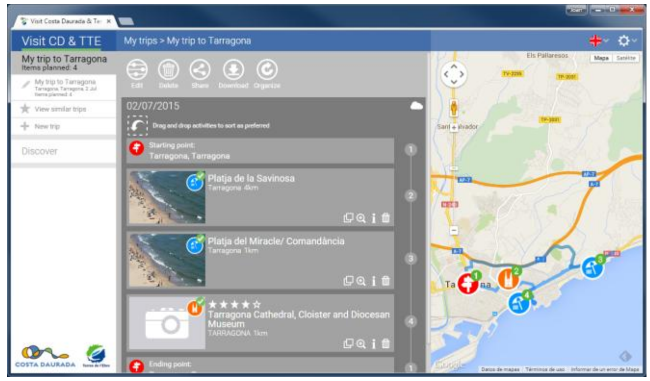


Figure 3. List and map of daily planned routes

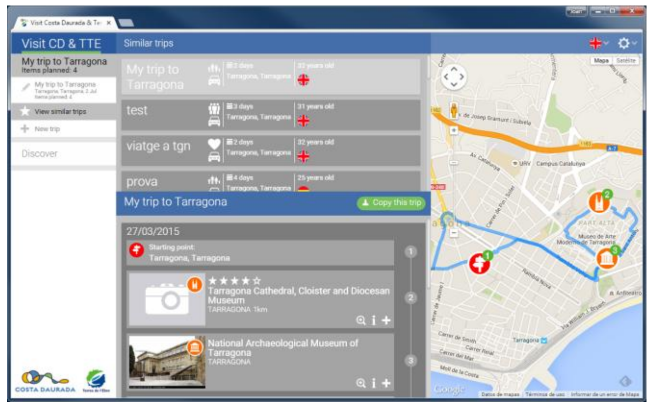


Figure 4. List of similar trips

Another social functionality added to the system permits sharing the trip with friends. There are two options to share. The private sharing allows sending a link by mail to other participants in the trip, so that they can access and modify the trip as they wish. The public option is to share the trip on social networks, such as Facebook or Twitter, in public mode. In this case all the user's friends will be able to see the trip, but they will not be able to modify it; however, they can copy it into another user session and then modify it.

2. REFERENCES

[1] Del Vasto-Terrientes, L., Valls, A., Zielniewicz, P., Borràs, J. (2015). A hierarchical multi-criteria sorting approach for recommender systems. Journal of Intelligent Information Systems, 1-34. DOI: 10.1007/s10844-015-0362-7.

[2] Borràs J., Valls A., Moreno A., Isern D. (2012) Ontology-based management of uncertain preferences in user profiles. Information Processing and Management of Uncertainty in Knowledge-Based Systems. Communications in Computer and Information Science (CCIS) 298, pp. 127-136. Springer-Verlag.

[3] Borràs J. (2015) Semantic Recommender Systems. Provision of personalised information about tourist activities. PhD Thesis. Universitat Rovira i Virgili, June 2015.

Finding More Fun for Travelers
Streamlined and Personalized Tourist Recommendations

Stacey Donohue
relEVENTcity
staceymdonohue@gmail.com

Sean MacLachlan
relEVENTcity
2301 Kootenai Street
Boise, Idaho 83705
sean@somewhere.us

Jeff Heath
relEVENTcity
jeff@biimail.com

ABSTRACT

In this demonstration, we present relEVENTcity, a mobile event recommendation application that matches individual tourists' preferences with local events in their destination city to create relevant, customized, and serendipitous event recommendations. Our app makes it easy to find great entertainment while traveling in unfamiliar cities.

Categories and Subject Descriptors
B.2.4. [**High Speed Arithmetic**]: Algorithms
H.3.3 [**Information Search and Retrieval**]: Information filtering, Relevance feedback

Keywords
Recommender systems, events, local, social network analysis.

1. INTRODUCTION

While other mobile applications have tried providing tourist recommendations, none have consistently provided relevant event recommendations to users. We solve this problem by creating an intuitive user interface that provides streamlined event recommendations personalized to each user.

Travel and lodging logistics often consume tourists pre-trip planning, which leaves little time for researching anything but the main tourist attractions at their destination. As a result, many travelers miss interesting events because they can not find the "right" events [1].

Several applications, including EventBrite (www.eventbrite.com), Eventseeker (www.evetntseeker.com), Yelp (www.yelp.com), Zofari [2] (which was acquired by Yahoo in 2014), and others have tried to solve this problem. But limited event data, and an emphasis on music and dining combined with overly-generalized recommendations have limited the event value of these apps to users. These recommendations have had reduced utility partly because they have not capitalized on the predictive value of linking online and offline social networks [3], social networks among individual users, and social influence between users [4].

In addressing these issues, we have developed relEVENTcity, a mobile application that generates *localized*, *customized*, and *serendipitous* event recommendations. relEVENTcity uses publically-available digital data to populate a complete event and venue taxonomy that uses a custom, hybrid recommender algorithm to match events with users' preferences. Recommendations are based on location, but users can choose alternate locations to make future trip planning easy. relEVENTcity pushes three event recommendations to each user daily based on that user's revealed preferences and location. Future plans include a web-based user interface, collaborative filtering from friend invites and attendance, and using more general popularity and population-based collaborative filtering to address the cold start problem for new users.

2. relEVENTcity

In this section, we provide a brief overview of relEVENTcity's recommendation engine.

2.1 Architecture

relEVENTcity uses Apache Spark with Scala as the framework to implement a custom hybrid recommender algorithm. Running in the Amazon Cloud, relEVENTcity's recommender connects to a database as a service using Orchestrate.IO in order to manage event, venue, performer, group and user data, and profiles.

2.2 Recommendation Algorithms

relEVENTcity uses a hybrid approach of a semantic ontology of events with a taxonomy of venues, combined with user collaborative filtering per city for generating recommendations. User preferences, actions and inactions build a user profile that is compared to an experience profile generated for events, based on event classification and venue taxonomy which enables us to make inferences about *why* a user

Figure 1. relEVENTcity's Welcome Screen

prefers particular events and recommend similar events based on a deeper modeling of the event experience, such as activity level, crowd size, formality, novelty, and other parameters. This enables

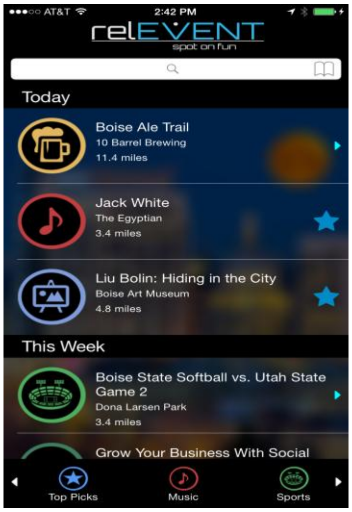


Figure 2. relEVENTcity's Daily Recommendations

Figure 3. relEVENTcity's Event Details

us to identify each user's hidden event preferences, which we then use to match each user to an experience, rather than just an event.

2.3 User Experience and Data Collection

Our goal is to simplify the user's event discovery process and most importantly, collect additional information about their preferences to provide ever-improving recommendations. Figure 1 shows relEVENTcity's welcome screen, which leads tourists through a series of questions to identify their interests, and provides the option to link with social media, which can provide constantly updated user interests with less direct effort from the user. Figure 2 shows relEVENTcity's daily, streamlined recommendations. With relEVENTcity, it only takes a few minutes for a tourist to learn about the most interesting events and find out if their friends have been there previously. Users can swipe right to learn more, swipe left to say "maybe later, but not today", and swipe down to indicate they are not interested in those types of events Figure 3 shows relEVENTcity's event screen. In addition to providing complete event details, it shows tourists if there is a conflict with their schedule, lets them add the event to their personal calendar, and lets them "follow" the venue or performer. In addition to transforming the user experience from passive to active, the "like," "dislike," "follow," and "add to calendar" options have been deliberately included to gather ongoing data about the user's preferences. This ongoing data stream is used to provide continually improving recommendations. Lastly, relEVENTcity includes a "discover" function which gives users the ability to unearth even more events in their traveling area.

3. VIDEO DEMONSTRATION

This video showcases relEVENTcity's sleek interface, the variety of events included, and the complete information provided about each event. The relEVENTcity video is available on YouTube at: https://www.youtube.com/watch?v=p9nVAqKXffQ&feature=youtu.be

4. INITIAL ASSESSMENT

The initial front-end prototype shows promising results and has received favorable feedback from industry experts and investors alike. While users have been pleased with the clean, playful, and intuitive app interface which is designed to be operated primarily with swiping motions rather than smaller buttons, there has been significant discussion about the optimal method for collecting user data in the initial set-up. Some users favor collecting that information with a brief series of questions while others believe that more complete and detailed information can be collected using radio buttons. Both options will be fully vetted as app development continues.

5. ACKNOWLEDGMENTS

Support for this project was provided by the Boise Metro Chamber of Commerce. Special thanks to Tony Hauser, George Johnson, and Faisal Shah for their entrepreneurial insight and advice.

6. REFERENCES

[1] de Macedo, A. and Marinho, L. Event Recommendation in Event-based Social Networks. In Hypertext, Social Personalization Workshop, p. 1,2014.

[2] http://techcrunch.com/2014/08/12

[3] Liu, X., He, Q., Tian, Y., Lee, W., McPherson, J., and Han, J. Event-based Social Networks: Linking the Online and Offline. In ACM KDD, p. 1032-1040, 2012.

[4] Ye, M., Liu, X., and Lee, W. Exploring Social Influence for Recommendation – A Probabilistic Generative Model Approach. ACM SIGIR, p. 671-680, 2011.

CT-Planner5: a Computer-Aided Tour Planning Service Which Profits Both Tourists and Destinations

Yohei Kurata
Tokyo Metropolitan University
1-1 Minami-osawa, Hahioji
192-0397, Japan
+81-42-877-2866
ykurata@tmu.ac.jp

Yasutaka Shinagawa
The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa
277-8568, Japan
+81-4-7136-4250
shinagawa@race.u-tokyo.ac.jp

Tatsunori Hara
The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa
277-8568, Japan
+81-4-7136-4250
hara_tatsu@race.u-tokyo.ac.jp

ABSTRACT
CT-Planner is a web-based tour planning service, which promotes the collaborative design of tour plans by a user and the system. Its remarkable feature is its interactive and cyclic process of tour planning, in which the user provides his/her requests in a stepwise manner and the system keeps revising the plan until he/she gets satisfied. Such interface design fits service domains where users are not well-aware of their interests/requests at the beginning, such as tour planning. This paper reports its latest version, CT-Planner5, which supports 25 destinations in Japan at this moment. CT-Planner5 makes it possible to generate tour plans which use public transportation when it is efficient. We explain the mechanism of plan generation, as well as the process of authoring destination data. We also introduce our future plan to use the user log of CT-Planner5 for destination marketing in collaboration with travel agencies, DMOs, and communities, as the collected user log allows us to know the demand and emerging trend of tourists for each destination without cost.

Categories and Subject Descriptors
H.3.5 **[Information Storage and retrieval]**: online Information – *web-based services*.

General Terms
Design, Human Factors.

Keywords
computer-aided tour planning, personalization, critiquing-based recommender system, user log, marketing analysis

1. INTRODUCTION
Making a tour plan is an exciting process of travel, but sometimes people feel hard to do it, especially when they are visiting unfamiliar destinations on a tight schedule. In order to relieve people from such difficulty, researchers have developed *tour recommenders* that generate personalized tour plans (e.g., [7][17][19]; See [26] for a review). Early systems typically aim at generating an optimal plan in a single step, asking users to give all tour conditions/requests in advance. This interface makes their users feel lack of participation [24]. In order to realize more user-centered planning, some systems introduced *customization phase*, in which the users are allowed to modify recommended plans [6][23]. Furthermore, we developed *CT-Planner* in pursuit for realizing system-user collaboration of tour planning [12]. In CT-Planner, the users can provide their requests in a stepwise manner and the system keeps revising the plan until they get satisfied. We confirmed that this interactive interface helps the users to clarify their requests and eventually leads to higher user satisfaction [15]. CT-Planner's interaction model is similar to that of *critiquing-based recommender systems* [5][18] in the sense that the users are requested to give feedbacks to the system about its recommendations. However, the requested feedback in CT-Planner is not the evaluation of each recommended plan, but pieces of requests which come up to the users' mind while examining the recommended plans.

This paper introduces the latest version of CT-Planner, called *CT-Planner5*. CT-Planner5 is available online at http://ctplanner.jp. CT-Planner5 succeeds the following features from its previous versions: collaborative planning approach, online accessibility, and use of a genetic algorithm for generating plans. In addition, CT-Planner5 now supports the generation of tour plans which use public transportation, and equips with two family tools, namely *plan viewer for smartphones* and *log analyzer*.

Our CT-Planner5 has two goals. The apparent goal is to provide a useful planning service to tourists on the Web. With CT-Planner5, people can consult on their plan from anywhere at any time, as much as they want. Another implicit goal is to collect log data from a large number of users, which can be eventually utilized as a resource for marketing analysis.

The remainder of this paper is structured as follows: Section 2 consider general requirements of computer-aided tour planning services. Section 3 looks back the history of CT-Planner. Section 4 describes the outline of CT-Planner5's service, including *plan viewer for smartphones*. Section 5 briefly explains how CT-Planner5 generates tour plans. Section 6 discusses the applicability of CT-Planner5's user log for marketing analysis, together with the introduction of our *log analyzer*'s prototype. Section 7 explains the process of data authoring in CT-Planner5. Finally, Section 8 concludes with a discussion of future work.

Workshop on Tourism Recommender Systems '15, September 20, 2015, Vienna, Austlia.

2. GENERAL REQUIREMENTS OF TOUR PLANNING SERVICES

Before introducing CT-Planner5, we discuss general requirements of tour planning service from a viewpoint of design engineering.

2.1 Design Perspective: Configuration Design by User

Thanks to advancement of ICT, the design and the use phases have become strongly and inseparably related. In some forms of design, users attend to the development process [3][22][28] and simultaneously design and use products/services. Many approaches have been used for achieving it, such as user-centered design [1], participatory design [21], lead-user innovation [10], and reinvention [4]. Hara *et al*. have introduced a model for combining design activities by the product/service provider, individual user, and user community [8]. They also explained engineering processes that involve and do not involve users. The importance of design by users is explained from the difference between these two processes. The discussion here does not distinguish between physical products and non-physical services.

Figure 1 shows such a general engineering process that does not involve users. The design process includes concept design and detailed design, and the production process includes production, manufacture, and quality control. In this entire process, users attend only to the use phase. In other words, users simply consume the products/services designed and produced by providers. In the design process, users only appear as the Voice of Customer (VoC), and their characteristics and behaviors are not considered.

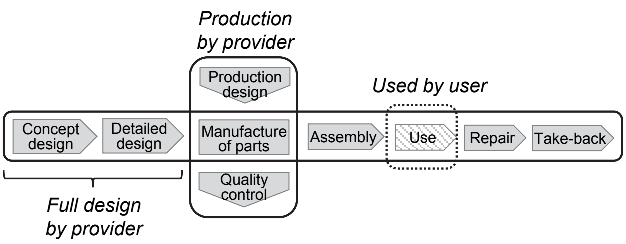


Figure 1. General engineering processes of a product/service (modified from [8])

On the other hand, Figure 2 shows an engineering process involving high user participation [8]. User participation is used as an approach to respond to each customer [22]. The main feature of this process is the processes of configuration and adaptation by the user. The process of full design by the provider is changed to that of preparatory design. In this process, providers design the basis of products/services and prepare for configuration by a user. In this situation, users not only use products/services but configure and adapt the parts prepared by the provider. Thus, they become co-designers of products/services. By providing users with more freedom for developing products/services, they can make these products/services more fit to their requirements and the conditions in the use phase. Preparatory design by the provider includes formulating the PFA (Product Family Architecture) [27] and preparing the configurator. Next, the process of configuration by the user is carried out as assembly and configuration in the configuration design of use. Interactive design methods have been proposed to support these user's processes (e.g., [29]). In the use phase, users use and adapt a product/service to the environment in the use phase.

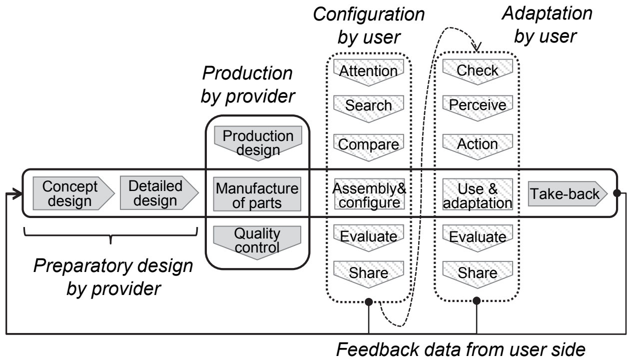


Figure 2. Engineering processes of a product/service involving high user participation (modified from [8])

2.2 Tour Planning Assistance

According to the above design perspective, tour planning is viewed as a configuration design activity of a service, in which tour components, mainly point of interests (*POIs*), are selected from numerous candidates and combined as into series, such that the plan maximizes the expected satisfaction of its users (i.e., tourists) under such constraints as total time, budget, start/goal point, mode of transportation.

Mathematically speaking, it is modeled as a combinatory optimization problem (Section 5). A unique feature of this design activity is that, not like product design, the design is conducted not only by experts, but often by ordinary users (in other words, people often make their tour plans by themselves). However, each user usually has insufficient knowledge about the destination and, accordingly, he/she is forced to estimate the value of each component and time necessary for enjoying it, as well as the travel time between these components, based on the limited amount of knowledge and experience. This task may be acceptable if the destination has only a small number of POIs within a walkable distance, but rapidly becomes difficult as the destination has larger number of POIs in a broader area. For instance, making the best one-day tour plan in Tokyo is an extremely difficult problem with countless number of possibilities. This justifies the significance of assistance service of tour planning by experts.

Of course, human experts, such as staffs at tourist information offices and hotel concierges, can assist individual tourists for tour planning. However, human resources of these experts are limited, especially during such busy hours as morning, and open hours of such services are also limited. In addition, people may feel reluctance to communicate with these human experts in foreign languages. Such situations motivate the development of computer-aided tour planning service, as it is expected to support tourists at any time at anywhere in the world without human cost.

Another unique feature of tour planning as a design activity is that users are often not well-aware of their own interests/requests at the beginning. This is simply because tourism is an activity that usually takes place at unfamiliar environments and people do not know what they can do at the destination. In this sense, tour planning should be distinguished from such online services as shortest-path finders and transit search tools where the users usually recognize their own interests/requests before use. Thus, we consider that tour planning service should have a certain *educational* aspect, providing appropriate amount of knowledge about the destination and promoting the users to find their own requests. Such knowledge enlightenment may also help the users at the destination if they want to rearrange their tour plan.

3. HISTORY OF CT-PLANNER

CT-Planner has a long history of improvement. It has its root on Kurata's preference-based tour planner [11], which asks its user to answer fifteen pairwise comparison questions to deduce his tour preference, and derives the best-scored personalized plan using a greedy optimization algorithm. In this system, however, the user is not allowed to modify the recommended plan. Thus, in order to achieve interactive tour planning, the first CT-Planner was developed [12]. The first CT-Planner allows such user requests as POI addition/removal. In addition, it has a *plan-comparing* function—it simultaneously shows two tour plans with different focuses, asks the user to select a preferable plan, and deduces his/her preference in a stepwise manner.

CT-Planner2 [13] has both a plan-comparing mode and a single-plan mode. In addition, it provides a manipulatable radar chart which represents preference model. In its user test, most users actually preferred the single-plan mode and manipulated his preference model by themselves. Thus, the plan-comparing function is no longer succeeded to the later versions.

CT-Planner became a web-based application since CT-Planner3 [14]. CT-Planner3 adopted a genetic algorithm for plan generation. Since we expected that this algorithm requires heavier computation than our previous greedy algorithm, we made two versions for experiment: a server-client version in which the plan generation is conducted on the server side, and a client-alone version. In our experiment, the response speed of the client-alone version was unexpectedly better than that of server-client version, even on mobile devices with low computation power. Thus, the client-alone model is succeeded to the later versions.

CT-Planner4 [15] supported multiple destinations and multiple languages. In order to increase the number of our destinations, we developed a macro-enhanced Excel template for authoring destination data.

In addition to the above versions, there were also several extended experimental versions of CT-Planner. For instance, Shimada *et al.* [25] revised CT-Planner4, such that it suggests the modification of user preference parameters based on his/her POI requests (Figure 3). Nakamura *et al.* [20] developed another extension of CT-Planner4, in which the user is proposed an alternative route in the last one-third part of his trip (Figure 4). This idea is motivated by our finding that people often deviate from their original plan made with CT-Planner, especially at the last part of their tour.

Figure 3. Proposal of modification of a user preference model based on his POI requests [25]

Figure 4. Proposal of an alternative route in the last part of the trip [20]

4. CT-PLANNER5

This section introduces the latest version of CT-Planner (CT-planner5) from a viewpoint of its user interface.

Figure 3 shows the initial screen of CT-Planner5. Here you are asked to select your destination and your favorite travel style. Currently CT-Planner supports 23 destinations in Japan, among which six has English contents as well. These destinations include i) large cities, which presumes the use of public transportations for sightseeing (e.g, *Sapporo*, *Yokohama*, *Nagoya*, *Kobe*, and *Fukuoka*), ii) subparts of Tokyo megalopolis, such as *Shinjuku* and *Shibuya*, which are typical units of a day trip in Tokyo, iii) walkable-size rural destinations such as *Shikine-jima Island* and *Ikaho Hot Springs*, and iv) even facility-scale destinations such as *Hongo Campus, University of Tokyo*. The favorite travel styles you can select are the following five: *anything welcome*, *city stroller*, *relaxed traveler*, *locality-oriented*, and *with children*. We adopted these five styles based on the result of our previous GPS-assisted survey on foreign tourists visiting Tokyo [2].

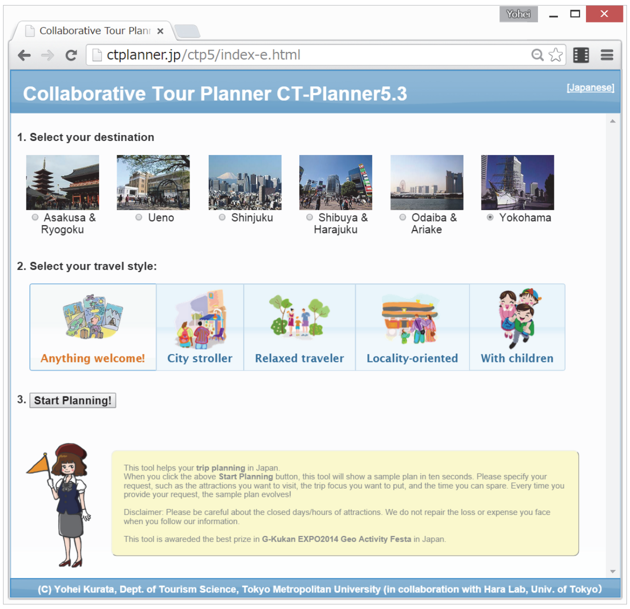


Figure 5. Initial screen (English version)

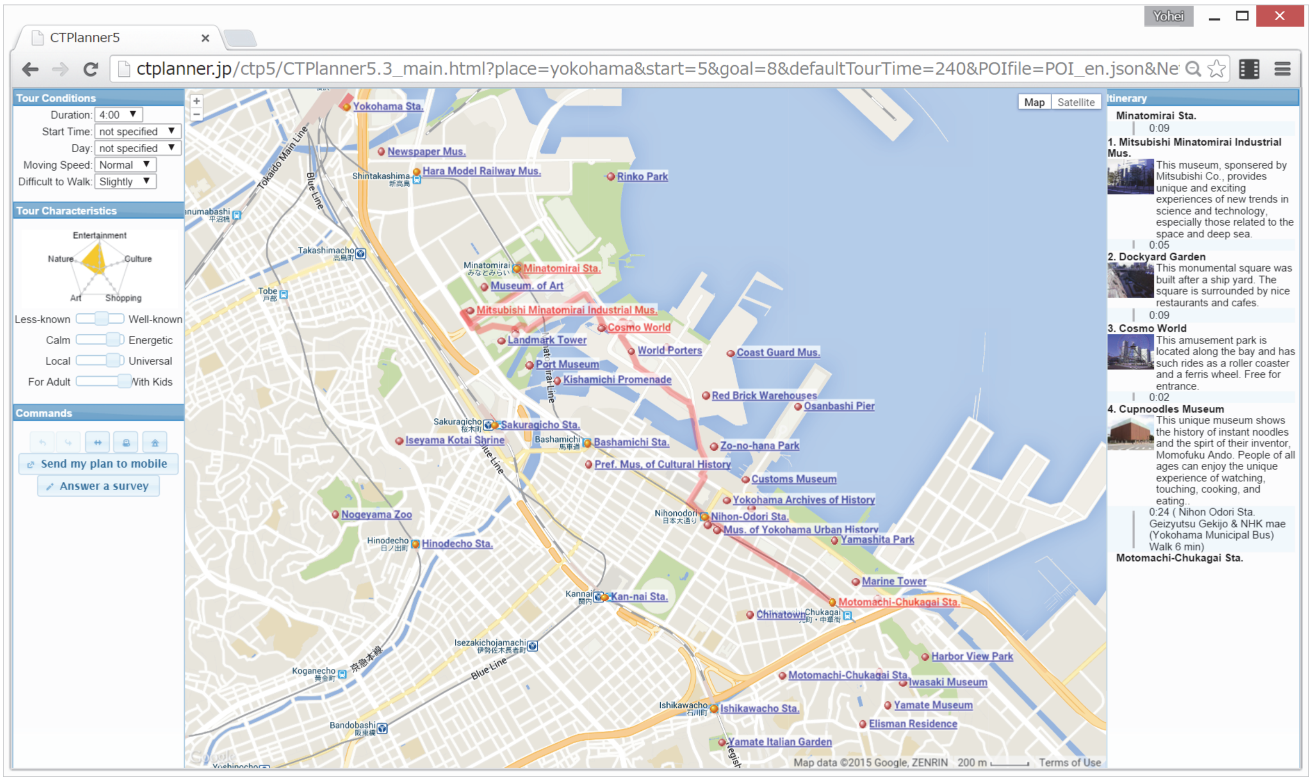


Figure 6. Main screen

Imagine that you select *Yokohama* and *With Children* as your destination and travel style, respectively. When you click the *Start Planning* button, you will see the main screen like Figure 6 in a few seconds. The main screen shows the route of an initial plan on the map, together with its itinerary on the right end. The initial plan is generated based on the travel style you have selected and typical tour conditions among Yokohama visitors (four-hour stay, starting from a west-side station, and ending at an east-side station).

The left end of the main screen shows your tour conditions and tour characteristics (see Figure 7 for detail). When you update any element on them, the displayed tour plan is revised immediately. The tour conditions consist of five items: *duration*, *start time*, *day of the week*, *walking speed*, and *degree of difficulty to walk*. The tour characteristics represent a model of user preference and consists of two parts: *focus* and *taste*. *Focus* refers to the tour's functional features that the user wants. It is represented by a manipulatable radar chart with five axes: *entertainment*, *culture*, *shopping*, *art*, and *nature*. If you put more weight on culture, for instance, your tour plan will visit more museums. *Taste* refers to the tour's emotional characteristics that the user wants. We considered four types of taste, each represented by two poles; *less-known* or *well-known*, *calm* or *energetic*, *local* or *universal*, and *for adult* or *with kids*. If you move the top slider to the right end (labeled *well-known*), for instance, your plan will visit popular places more likely. Note that the initial value of each parameter is determined based on your initial selection of favorite travel styles, such that the user do not worry about the settings of these parameters at the beginning. These initial values are predetermined by us.

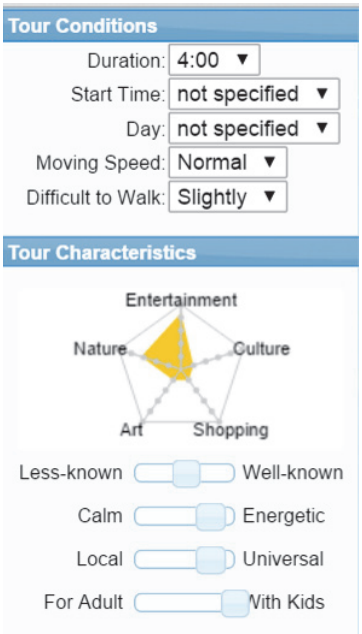


Figure 7. Tour conditions and characteristics

The map shown on the center (Figure 6) is depicted with the aid of Google Maps JavaScript API. Accordingly, you can zoom/scroll the map and even see the isometric view of satellite images, which is nice to understand the tour route (Figure 8). If you click the name tag of a POI on the map or the satellite image, an information window opens (Figure 9). This window shows the

POI's name, estimated value for the user, description, hyperlinks to the related websites (if available), open hours/days, staying time, photo, scores on the five items, and several buttons. If you click *Visit* button, the system generates tour plans which visit this POI as long as possible. Conversely, once you click *Avoid* button, the system no longer shows the plans that visit this POI. *Start*/*Goal* button allows you to set this POI as the tour's start/goal location. Finally, *+10*/*-10* button allows you to adjust the staying time of this POI. Note that CT-Planner5 does not force the users to select all POIs they want to visit—instead, they can entrust the selection of POIs to the system other than those with their visit requests.

Figure 8. The route of a recommended tour plan displayed over an isometric satellite image

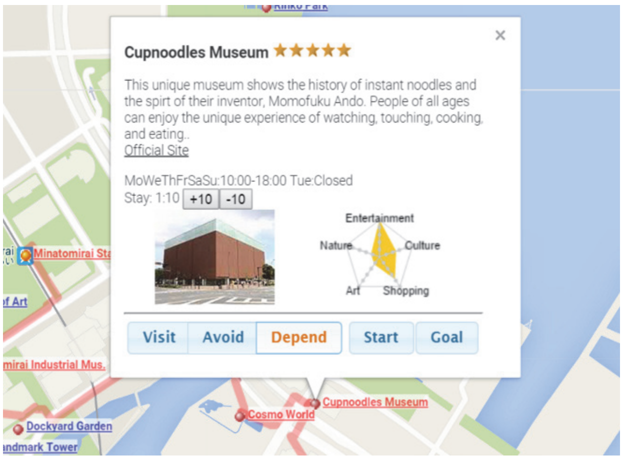


Figure 9. An example of information window

The design of CT-Planner5 presumes the following interaction cycle between the user and the system:

1. the user examines the displayed tour plan, as well as the POIs on/off the route,
2. the user gives certain feedback to the system (modification of tour conditions, modification of tour characteristics, or visit/avoid request), and
3. the system revises the plan accordingly and displays it.

This cycle is repeated until the user gets satisfied with the displayed plan. This cyclic interface relieves its user from the burden of specifying their requests at the beginning of planning [12].

When you get satisfied with your plan, you can print it or send it to your smartphone via a two-dimensional barcode (Figure 10). Our *plan viewer for smartphones* shows your plan and your current location on the screen. In addition, this viewer can show you the latest transportation schedule with the aid of Yahoo Transit.

Figure 10. Linkage to our plan viewer for smartphones

5. UNDERLYING MECHANISM

CT-Planner5's mechanism for generating tour plans is similar to that of CT-Planner4 [15]. Here we briefly describe its essence. Basically, CT-Planner5 estimates the value of individual POIs for each user, and calculates the most efficient plan under given constraints that maximizes the sum of the estimated values of the POIs to be visited in the tour.

The value of each POI is estimated from the matching between the POI's characteristics and the current setting of the tour characteristics that the user wants. The previous versions of CT-Planner considers simple weighted sum of the scores in five *focus* categories (i.e., *culture*, *entertainment*, *nature*, *art*, and *shopping*), but this method is advantageous to the POIs with balanced scores. Thus, CT-Planner5 modifies this method, such that the POI with a high score in a specific category is highly evaluated if it matches with the user's interest. Note that the estimated value of a POI is replaced to *zero* if the user wants to avoid this POI, and to a very high value if he wants to visit it.

The calculation of the most efficient plan under given time and start/goal constraints is formalized as a selective traveling salesman problem [16]. Since this is a NP-hard combinatory optimization problem, we adopt a genetic algorithm (GA) for deriving semi-optimal solutions in a short time. In this algorithm, each tour plan is regarded as a gene (i.e., a series of symbols, each representing a POI). We randomly generate thousands of initial genes and then simulate evolution (crossover and mutation) and survival competition over a large number of generations. In each competition, the plans with higher evaluation survive more likely (but not always). This process eventually leads to the generation of superior plans, although the optimality of the final plan is not guaranteed. A nice point of this algorithm is that we can easily support open/close hours of POIs, and even temporally-changing values of POIs, only by adjusting the evaluation function of each plan.

Of course, the above method still has a room for improvement. In reality, the plan's value may be affected by the combination and order of POIs to be visited. We should also consider reciprocal effects—if people visit similar POIs repeatedly, they may get bored, or conversely, they may get more satisfaction. How to incorporate such effects into the model is our future question.

6. USE OF LOG DATA

CT-Planner5 records all operations and plans by each user. This log will be useful for navigating the same user at the destination. Our previous experiment shows that our users often deviate from the plan they have made, because the interaction with CT-Planner helps them to identify the interesting POIs near their route. This implies that not only the final plan they have made, but also the list of the POIs they have checked are useful for a smart on-site navigation.

The user log of CT-Planner5 is also useful from a viewpoint of destination marketing, if it is analyzed statistically. We are developing *CT-Planner5 log analyzer* powered by Google Analytics (Figure 11) to examine the following statistics:

- the number of accesses,
- average tour durations,
- viewing rate of each POI (i.e., how much percentage of users have opened its info-window),
- adopting rate of each POI (i.e., how much percentage of users have made the tour plans that visit it), and
- appearance of POI pairs (i.e., which POIs are often listed together in user-generated tour plans),

as well as conducting data-mining analyses for discovering unique tour plans. A nice point is that we can examine such statistics for each user group (genders, ages, countries, languages, and interests) and compare them with the aid of Google Analytics. Such information will help DMOs to examine their promotion strategies, as well as travel agencies to design their package tours [9]. Another nice point is that, not like ordinary questionnaire-based tourists surveys, we do not have to pay any extra cost to collect data from customers.

Figure 11. Prototype of CT-Planner5 Log Analyzer

7. AUTHORING DESTINATION DATA

We made a macro-enhanced Excel template with which people can easily make a destination data for CT-Planner (Figure 12). On this template the user input for each POI the name, description, type, URLs of related websites, URL of photo data, open days/hours, and scores for the nine criteria (Section 4). It normally takes a few hours to complete the table for a typical destination with thirty to sixty POIs.

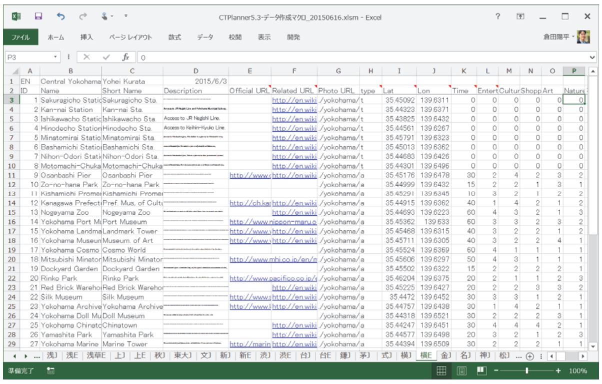


Figure 12. CT-Planner5's template for destination data

Once the table is completed, its macro program computes the route between every pair of the POIs listed in the table, in order for CT-Planner5's main program to reduce the time for generating tour plans. Our template has several modes for the route calculation. In *walking-only mode* and *driving mode*, the user can select either Google Directions API or Map Quest API (using OpenStreetMap data) as a route calculating engine, considering the coverage area of these two services. In *public transportation mode*, the program uses both Google Directions API and Yahoo Transit, because Google Directions API does not provide the routes that use public transportations in Japan, while Japanese version of Yahoo Transit provides such routes together with their average travel time, but not with their geometrical shapes. In *public transportation mode*, the walking route between a pair of POIs is calculated only if the straight distance between them is less than 1.5 km, while the possibility of public transportation use is sought if the distance is more than 1km. By this mechanism, the number of API requests is considerably reduced. Normally this route calculation process takes about ten minute to one hour.

We have tested the above macro-enhanced template with several groups, including students, professionals, and local people. Through the collaboration with them, we learnt that the involvement of various people in data creation is effective not only for expanding CT-Planner's coverage, but also for enriching the diversity of our service. We are, therefore, seeking further possibility of user participation, by introducing a web-based *CT-Planner's data editor*, with which people can create destination data in a collaborative manner.

8. CONCLUSION AND FUTURE WORK

This paper reports the latest version of our web-based tour planning service, CT-Planner5. It promotes the collaborative design of tour plans by the user and the system through its cyclic interaction. In addition, this system now equips with sub-systems for viewing the plans for smartphones, and analyzing user logs.

In Japan, the number of tourists from abroad increases rapidly due to the nation's policy toward a tourism-oriented country, as well as rapid economic growth in East Asia. The interests of visitors to Japan are much more diversified than those of domestic tourists who have dominated our market for long. As a result, Japanese tourism industry is facing two critical problems: how to know the interest of inbound visitors and how to serve them appropriately and efficiently based on such knowledge. We expect that CT-Planner will contribute to the tourism industry under this situation. Firstly, CT-Planner serves as a virtual consultant of tour planning.

Since CT-Planner is an online service, it allows inbound tourists to consult their tour plans at any time at anywhere, without paying human cost. Secondly, CT-Planner will serve as a survey tool, with which we can know the interest of inbound tourists. In order to achieve these two goals, we are now working on the translation of contents into English, Chinese, and Korean languages, as well as the enrichment of destinations supported by CT-Planner5.

Figure 13 shows our future vision. We expect that CT-Planner basically serves individual tourists who are relatively advanced in the sense that they actively collect tourist information by themselves. In exchange for providing planning service to them, we will obtain their planning log, from which we can analyze their demand and emerging trend. We will provide the result of our analyses to travel agencies and DMOs. This information will be useful for providing more attractive tour merchandise and/or information service to tourists in general. This will motivate those organizations to cooperate with us and provide their destination data. To realize this virtuous cycle is our future challenge.

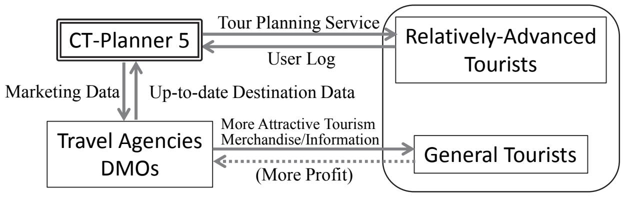


Figure 13. Future vision of CT-Planner

Furthermore, we envision the application of CT-Planner to tourism-oriented community design. Tourism is a key to achieve a sustainable community development in both economic and cultural aspects. Meanwhile, it has been difficult for local people to know how visitors evaluate their region and what the community should improve it from the viewpoint of tourism. Now the user log of CT-Planner, as well as user-posted contents in such SNSs as Trip Advisor, Twitter, and Flickr, are available for the visualization of visitors' activities and evaluations. Such visualization will serve as a useful material for the local people to discuss the future of their community by themselves. In order to demonstrate its potential, we are currently working with several local communities in Japan, where we have created CT-Planner's destination data with local people, started planning service online (and eventually offline at information kiosks), and kept collecting the log data from various users for future analysis.

Another future challenge for us is to develop a smart on-site navigation tool, which works in combination with CT-Planner. We consider not a virtual tour conductor, as seen in [19], which controls the user to follow his tour plan, but the one that supports his flexible trip, taking his original plan and planning activity into account. Then, by analyzing the difference between the original plan and the actual behaviors recorded by this tool, we want to pursue deeper understanding of tourists.

9. ACKNOWLEDGMENTS

This work is supported by JST RISTEX's Implementation Support Program (Project Title: Platform Foundation and Social Implementation of Computer-aided Tour Planning Technologies That Contribute the Harmony of Tourists and Communities).

10. REFERENCES

[1] Abras, C., Maloney-Krichmar, D., Preece, J., 2004. User-Centered Design, *Encyclopedia of Human-Computer Interaction*, Thousand Oaks: Sage Publications.

[2] Aratani, K., Shimada, S., Ota, J. and Hara, T. 2012. Classification of inbound tourist activities using GPS log data toward service innovation. *12th Japan-Korea Design Engineering Workshop*, 28-31

[3] Barki, H., Hartwick, J. 1994. User Participation, Conflict, and Conflict Resolution: The Mediating Roles of Influence. *Information System Research*, 5, 4, 422-4384.

[4] Carroll, J. 2004. Completing Design in Use: Closing the Appropriation Cycle. *European Conference on Information Systems* (Turku, Finland, June 14-16, 2004). CD-ROM.

[5] Chen, L., and Pu, P. 2009. Interaction Design Guidelines on Critiquing-Based Recommender Systems. *User Modeling and User-Adapted Interaction*, 19, 3, 167-206.

[6] Garcia, A., Arbelaitz, O., Linaza, M., Vansteenwegen, P., and Souffriau, W. 2010. Personalized Tourist Route Generation. *Current Trends in Web Engineering* (Vienna, Austria, July 5-6, 2010), CWE 2010 Workshops, Lecture Notes in Computer Science 6385, 486-497.

[7] Goy, A., and Magro, D. 2004. STAR: a Smart Tourist Agenda Recommender. *Workshop on Configuration* (Valencia, Spain, Aug. 2004). *ECAI 2004*, 8/1-8/7

[8] Hara, T., and Arai, T. 2013. Design-of-use and design-in-use by customers in differentiating value creation. *CIRP Annals - Manufacturing Technology*, 62/1 (ISSN 1660-2773), 103-106.

[9] Hara, T., Shimada, S., Yabe, N., Kurata, Y., Aoyama, K. and Hompo, Y. 2012. Value Co-Creation in Tourism: Incorporating Non-Expert's Design into Expert's Design Activities. *1st International Conference on Human Side of Service Engineering* (San Francisco, CA, July 21-15, 2012).

[10] Hippel, V. E. 2005. *Democratizing Innovation*, London: MIT Press.

[11] Kurata, Y., Sadahiro, Y., and Okunuki, K. 2000. Development of a Preference-Based Tour Planning System. *9th Conf. of GISA Japan* (Tokyo, Japan, Oct 2000). 199-202 (in Japanese).

[12] Kurata, Y. 2010. Interactive Assistance for Tour Planning. *Spatial Cognition VII* (Mt. Hood, OR, Aug. 15-19, 2010). Lecture Notes in Artificial Intelligence 6222, 289-302.

[13] Kurata, Y. 2011. CT-Planner2: More Flexible and Interactive Assistance for Day Tour Planning. *ENTER 2011* (Innsbruck, Austria, Jan. 26-28,2011). 25-37.

[14] Kurata, Y. 2012. CT-Planer 3: An Interactive Support Tool for Tour Planning on the Web. *International Journal of Tourism Science* 5, 159-165 (in Japanese).

[15] Kurata, Y. and Hara, T. 2014. CT-Planner4: Toward a More User-Friendly Interactive Day-Tour Planner. *ENTER 2014* (Dublin, Ireland, Jan. 21-24, 2014), 73-86, Dublin, Ireland.

[16] Laporte, G., and Martello, S. (1990). The Selective Travelling Salesman Problem. *Discrete Applied Mathematics*, 26, 2, 193-207.

[17] Lee, J., Kang, E., and Park, G. 2007. Design and Implementation of a Tour Planning System for Telematics Users. *ICCSA 2007* (Kuala Lumpur, Malaysia, Aug. 26-29, 2007). Lecture Notes in Computer Science 4707, 179-189.

[18] Linden, G., Hanks, S., and Lesh, N. 1997. Interactive Assessment of User Preference Models: The Automated Travel Assistant. *6th Intl. Conf. on User Modelling* (Chia Laguna, Sardinia, Italy, June 2-5, 1997), 67-78.

[19] Maruyama, A., Shibata, N., Murata, Y., Yasumoto, K., and Ito, M. 2004. A Personal Tourism Navigation System to Support Traveling Multiple Destinations with Time Restrictions. *Advanced Information Networking and Applications 2004* (Fukuoka, Japan, Mar. 29-31, 2004). 2, 18-21.

[20] Nakamura, Y., Shimada, S., Ogata, T., Kurata, Y., Ota, J. and Hara, T. 2014. An Interactive Tool for Tour Planning Considering Adaptive Behaviors. *JSPE Spring Conference 2014* (Tokyo, Japan, Mar. 18-20, 2014). CD-ROM (in Japanese).

[21] Pilemalm, S., Timpka, T. 2008. Third generation participatory design in health informatics--making user participation applicable to large-scale information system projects, *Journal of Biomedical Informatics*, doi:10.1016/j.jbi.2007.09.004, April 2008.

[22] Robey, D., Farrow, D., 1982. User Involvement in Information System Development: A Conflict Model and Empirical Test. *Management Science*, 28, 1, 73-85.

[23] Schaller, R. 2011. Planning and Navigational Assistance for Distributed Events. *2nd Workshop on Context Aware Intelligent Assistance* (Berlin, Germany, Oct. 2011) 44-56.

[24] Seifert, I. 2008. Collaborative Assistance with Spatio-temporal Planning Problems. *Spatial Cognition V* (Freiburg, Germany, Sep. 15-19, 2008). Lecture Notes in Computer Science 4387, 90-106.

[25] Shimada, S., Ogata, T., Ota, J., Hara, T., and Kurata, Y. 2014. A Support for Design of Use in Consideration of Use Phase. *13th International Design Conference* (Dubrovnik, Croatia, May 19-22, 2014). 995-1004.

[26] Souffriau, W. and Vansteenwegen, P. (2010) Tourist Trip Planning Functionalities: State-of-the-Art and Fitire. *Current Trends in Web Engineering* (Vienna, Austria, July 5-6, 2010). ICWE 2010 Workshops, Lecture Notes in Computer Science 6385, 474-485.

[27] Tseng, M. M., J. Jiao and M. E. Merchant 1996. Design for Mass Customization. *CIRP Annals - Manufacturing Technology*, 45, 1, 153-156.

[28] Yang, B., and Burns, N. 2003. Implications of postponement for the supply chain. *International Journal of Production Research*, 41, 9, 2075-2090.

[29] Yanagisawa, H., Fukuda, S., 2003, Interactive Design Support System by Customer Evaluation and Genetic Evolution: Application to Eye Glass Frame. *7th International Conference on Knowledge-Based Intelligent Information & Engineering Systems* (Oxford, UK, Sep. 3-5, 2003), Lecture Notes in Computer Science 2774, 481-487.

The importance of Diversity in Profile-based recommendations: A Case Study in Tourism

Fernando Sanchez-Vilas
CITIUS
University of Santiago de Compostela
Santiago de Compostela, Spain
fsvilas@gmail.com

Jasur Ismoilov
CITIUS
University of Santiago de Compostela
Santiago de Compostela, Spain
njasur@yahoo.com

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

ABSTRACT
The paper explores the concept of similarity between two users measured in user profile space rather than the traditional rating space. The study aims at discovering the most relevant user profiles in order to provide recommendations to any given target profile. Closer profiles were found to be the most accurate in terms of prediction error. However, the best results were obtained when including profiles far away from the target one. This striking result is explained in light of the diversity prediction theorem.

Keywords
User profile, Diversity Prediction Theorem, tourism, gastronomy.

1. INTRODUCTION
A search on a tourism website typically involves a user filling a form to choose the item of interest (hotel, travel package and so on) as well as the relevant dates of the trip. The response usually includes prices and availability of those items fulfilling the request, which actually assists the consumer to make an informed decision. In the Tourism 2.0 era, the decision-making process has been further facilitated by means of specialized websites that include additional feedback provided by travelers who previously experienced the evaluated item. This feedback comes under different flavors (reviews, ratings and comments) and serves to further clarify the quality of services or to uncover issues or problematic situations. However, the relevancy of this additional experience-based information is not the same for all future users. Recently, some advanced websites started to pave the way for personalization services by classifying user feedback on the basis of user profiles, like Families, Couples and Business profiles. The motivation for this move comes from the fact that a touristic product could be a wonderful experience for, let's say, a Family, but not appropriate for a Couple. The perceived trend is to increase the tourist satisfaction by providing better personalization services on the basis of relevant consumer attributes, both personal and contextual [1].

A number of Recommender strategies based on consumer attributes have been proposed in the literature. They can be classified according the scheme followed to generate the predictions: (1) probabilistic predictors, and (2) rule predictors. Among the first ones, an algorithm proposed by Ono et al. (2007) uses contextual information including user, item and context attributes, to recommend movies. The interaction scenario is modeled by: (1) a set of user profile attributes (U), with attributes like age and sex, (2) a set of context attributes (S), with attributes like mood and location, (3) a set of item attributes (C) with attributes like film genre and director, (4) a set of film ratings enhanced with contextual information. Their approach applies Bayesian networks to obtain the probability $P(V|u, s, c)$ of a rating for the target user $U = u$, specific context $S = s$ and candidate movie $C = c$. The Bayesian paradigm is also used in another recommender system but pointing out a difference between a fixed profile stated by the user, and an adaptive profile that is built dynamically based on user activity [3]. A naive Bayes network was also applied to generate recommendations adapted to contexts that were previously predicted by means of behavioral pattern analysis [4]. In the field of tourism, Costa et al. (2012) propose a probabilistic classifier and an agent-based approach in which a set of user attributes, a user context attributes and some item context attributes are defined: the restaurant category, the price, the schedule, the kind of day, the distance, the timeOfDay, the day of the week and the goal of the user interaction. The recommendations are conditioned by the users goal on any given moment. Among the second ones, a restaurant recommender system made up with a set of semantic rules was proposed by Vargas et al. (2011). The rules depend on user properties and context information. The key aspect of this work is the selection of the most relevant context attributes among an original set of 23 restaurant attributes, 21 user attributes and 2 environmental attributes.

In this paper we aim at exploring the value added by user attributes under the traditional k-Nearest Neighbors (k-NN)

*To whom correspondence should be addressed.

Figure 1: TapasPassport of Santiago(é)Tapas contest. The official information of the contest, including the set of available tapas as well as the location of the restaurants, was published in the TapasPassport.

scheme [7, 8]. User attributes are used to characterize user profiles, and then k-NN predictors can work with a similarity measure built on top of a user profile space rather than a rating space. In what follows, we present the hypothesis underlying the research work, the experiment carried out to test the hypothesis, the exploratory analysis that provided the key findings to develop the profile-based algorithms, the evaluation of those algorithms, and finally, the application of the diversity prediction theorem to understand an striking result obtained with our algorithms.

2. HYPOTHESIS

The line of thought behind this work is that k-Nearest Neighbors predictors in the user profile space, i.e user attribute space, should have to work better than in rating space. The rationale is that the similarity metrics behind the traditional user-based approach, a k-NN predictor in rating space, is not really measuring the similarity in tastes between two users. In other words, the fact that two users present the same set of ratings on a number of items does not mean that the two share the same tastes on those items. A coincidence on ratings does not imply a coincidence on tastes. Ratings measure satisfaction, an outcome of the experience process, while tastes measure preference on the attribute space of a choice set. In short, we believe that the probability of two users having similar tastes is higher when the two are closer in profile space rather than in Rating space.

3. EXPERIMENTAL DESIGN

3.1 Santiago(é)Tapas contest

In the context of the RECTUR project, an experiment was carried out with real users in the context of Santiago(é)Tapas, a gastronomic contest 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 (Fig. 1). 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 by covering the vote shown in Fig. 2. 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

Figure 2: Santiago(e)Tapas Vote. The participants had to fill the Tapa Votes with both the tapa and the overall experience rating.

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 on a given moment determines an *Interaction* of a *User* with a *Tapa* that will elicit a satisfaction response quantified as a user *Rating*. Table 1 shows some relevant figures of the experiment.

In order to avoid the overload of contest participants with a large list of feedback questions, only a set of attributes of the full research model has been included in the evaluation process. These attributes were selected with the help of experts in the field of gastronomy. Figure 2 shows the tapa vote that was finally designed to gather the experience of the user after a tapa consumption (user-tapa interaction).

For each tapa, we gathered the following attributes:

- Type: Meat, Fish, Vegetables, etc. 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, atmosphere and style.
- Average Rating. The average of ratings provided by consumers.

The consumers, in turn, were characterized with the following attributes:

- Origin: There will be differences between local, Spanish and foreign users as the first ones have a deeper knowledge of both restaurants and gastronomy.
- Character: Users are classified either as daring or as traditional based on the tapas they have consumed. A group of experts grouped the tapas offered in the contest in two groups, traditional and daring. Experts considered several attributes like tapa ingredients or tapa presentation in order to classify them.
- Experience: User domain knowledge will increase owing to the consumption of new tapas. Due to this, it was assumed that user ratings accuracy will increase with the experience she has.

At each tapa consumption, the user had a user context that was described as follows:

- Social context defined by the user company
- Temporal context defined by the time frame when she consumes the tapa, the hour, the day, the kind of day (work day or holiday)
- Climatological context defined by weather conditions
- Location context defined by the position of the user when she decides to start a tapa consumption.

Table 1: Experiment Info

Participating restaurants	56
Different tapas offered	109
Tapas consumed	35.000

4. EXPLORATORY ANALYSIS

4.1 Methods

Exploratory data analysis was developed by John Tukey in the field of Statistics to encourage researchers to explore the data in some informal way in order to discover patterns or relationships between different variables [9]. The idea behind this approach is to generate hypothesis that could lead to further experiments and/or specific confirmatory analysis.

We found this strategy suitable to explore our idea about the usefulness of user profile information to generate better recommendations. On the basis of the available user attributes and attribute levels (see Table 2 for details), 18 profiles were identified and every user of the Rectur Dataset were categorized on each one of those profiles. Thereafter, a number of target users as well as their collection of tapa ratings was chosen. For each tapa rating, a prediction was generated by averaging just the ratings of those users in profile k (k ranging from 1 to 18)). The Mean Absolute Error (MAE) was used to test the accuracy of the prediction according to each profile k. MAE is estimated by comparison of the prediction with the real rating value in the following way:

$$MAE = \frac{1}{n_k} \sum_{i=1}^{n_k} |\hat{r}_i - r_i| \qquad (1)$$

where n_k is the number of users in profile k, $\hat{r}_i$ is the predicted rating, and r_i is the real rating provided by the target user.

4.2 Results

The results of the exploratory analysis are shown in table 3. For brevity, only five target user profiles are presented. For each target profile, the best as well as the worst predictor in terms of MAE are presented. From this sample it can be observed that the best predictors correspond to those that are closer to the target profile according to their attribute values. There seems to be a correlation between the accuracy of the prediction and the similarity between profiles in the user attribute space. This finding motivated the development and evaluation of profile-based algorithms.

5. PROFILE-BASED ALGORITHM

5.1 Methods

The similarity between user profiles is determined by means of a distance measure in the user profile space. The contribution of each rating to the final prediction is weighted using this profile similarity among the active user and the target user instead of the traditional rating similarity between users.

Table 2: Attributes and Levels of User Profiles

Attributes	Levels
Character	Traditional, Daring
Origin	Local, Spanish, Foreign
Experience	Low, Medium, High

Table 3: Exploratory Analysis: MAE per user profile. Abbreviations stand for: DAR (Daring), TRA (Traditional, FOR (Foreign), SPA (Spanish), MED (Medium).

Target	Predictor	MAE
DAR,FOR,HIGH	Best: DAR,FOR,HIGH	0,50
	Worst: TRA,FOR,LOW	1,21
DAR,SPA,MED	Best: DAR,SPA,MED	0,00
	Worst: DAR,FOR,HIGH	1,50
TRA,FOR,HIGH	Best: DAR,FOR,HIGH	0,67
	Worst: DAR,SPA,MED	1,50
TRA,FOR,MED	Best: DAR,FOR,MED	0,50
	Worst: TRA,SPA,HIGH	1,66
TRA,SPA,MED	Best: TRA,SPA,HIGH	0,33
	Worst: TRA,FOR,MED	1,25

A metric of profile distance has been defined on the basis of the distances between profile attributes, which are shown in Tables 4, 5 and 6. To compute the profile distance between a user with profile i and a user with profile j, the following equation was used:

$$d_{i,j} = \sum_{a \in A} d^a_{i,j} \qquad (2)$$

where A is the set of attributes and $d^a_{i,j}$ are the distances between profiles i and j regarding to attribute a . Finally, the similarity between profiles is calculated as follows:

$$s_{i,j} = \frac{1}{1 + d_{i,j}} \qquad (3)$$

where $d_{i,j}$ the distance between profile i and j. When $d_{i,j} = 0$, the similarity $s_{i,j} = 1$, the maximum value. Similarity then decreases as long as the distance between profiles increases.

The prediction was estimated in two different ways: (1) basic weighted average of neighboring users ratings (equation 4) , and (2) compensated weighted average of neighboring users ratings (equation 5) [10]. The equations for both schemes are:

$$\hat{r}_{j,k,l} = \frac{\sum_{i \in profile_k} s_{j,i} \times r_{i,k,l}}{\sum_{i \in profile_k} s_{j,i}} \qquad (4)$$

Table 4: Distances between Character values.

u_k/u_l	Daring	Traditional
Daring	0.0	1.0
Traditional	1.0	0.0

Table 5: Distances between Origin values.

u_k/u_l	Local	Spanish	Foreign
Local	0.0	1.0	2.0
Spanish	1.0	0.0	1.0
Foreign	2.0	1.0	1.0

$$\hat{r}_{j,k,l} = \overline{r_l} + \frac{\sum_{i \in profile_k} s_{j,i} \times (r_{i,k,l} - \overline{r_i})}{\sum_{i \in profile_k} s_{j,i}} \tag{5}$$

where $\hat{r}_{j,k,l}$ is the predicted rating for user j in profile k for item l, $s_{j,i}$ the similarity computed under equation 3, $r_{i,k,l}$ the rating of user i of profile k on item l, and $\overline{r_i}$ the average of ratings of user i.

5.2 Results

The first analysis was focused on estimating the MAE for predictors based on all user profiles at the same distance $d_{i,j}$ from target user i. Results with increasing distances from the target user are shown in table 7 and plotted in figure 3. It is observed that lower MAEs correspond to closer distances, i.e. higher similarities according to equation 2. The error increases with distance until reaching its maximum value. These results confirm the exploratory analysis performed in the previous section. However, the traditional user-based algorithm still outperforms the best profile-based predictor, the one with user profile at distance $d = 1$ (see Table 10).

The second analysis was aimed at analyzing the impact of aggregating the users at different distance profiles. In short, we have generated the predictions at distance d with all users in profiles with distances lower or equal to d. The results of such aggregation profile scheme is shown in Table 8. A striking pattern is found here, as the MAE decreases when increasing the profile distance. This behavior was confirmed under the compensated weighting prediction of equation 5 (results in Table 9). However, the consequence of this outcome is that Profile-based algorithms with aggregation and compensated weighting prediction could work slightly better than traditional user-based approaches (see Table 10).

The question now opened is how to explain the fact that aggregating lower accurate predictors, i.e. those with higher distance and independent higher MAE, results on a decrease in MAE. The next section is focused on answering this question.

6. DIVERSITY PREDICTION THEOREM

Lu Hong and Scott Page propose what they called Diversity Prediction Theorem which states that the squared error of a collective prediction equals the average squared error of individual predictions minus the predictive diversity [11]. This

Table 6: Distances between Experiences values.

u_k/u_l	Low	Medium	High
Low	0.0	1.0	2.0
Medium	1.0	0.0	2.0
High	2.0	1.0	0.0

Table 7: MAE Results for predictors based on user profiles with increasing distances: basic weighting prediction.

Distance	MAE	Average Predictors
0	0.87236	16.30981
1	0.86204	20.56839
2	0.86851	16.14271
3	0.93652	7.27814
4	1.02836	3.03665
5	1.03636	2.33515

Table 8: MAE Results for predictors based on aggregation of user profiles with increasing distances: basic weighting prediction.

Distance	MAE	Average Predictors
0	0.87236	16.30981
0+1	0.8424	35.9854
0+1+2	0.83232	51.86515
0+...+3	0.83149	58.70979
0+...+4	0.83132	60.21792
0+...+5	0.83112	60.59597

Table 9: MAE Results for predictors based on aggregation of user profiles with increasing distances: compensated weighting prediction.

Distance	MAE	Average Predictors
0	0.7981	16.30981
0+1	0.78246	35.9854
0+1+2	0.77478	51.86515
0+...+3	0.77316	58.70979
0+...+4	0.77296	60.21792
0+...+5	0.77288	60.59597

Table 10: Comparison between profile-based and user-based algorithms.

Algorithm	MAE
Profile-based (Best predictor at d=1)	0.86
Profile-based (Aggr. + Basic Weight)	0.83
Profile-based (Aggr. + Comp. Weight)	0.77
User-based(k=500,minxy=3,x=3)	0.78

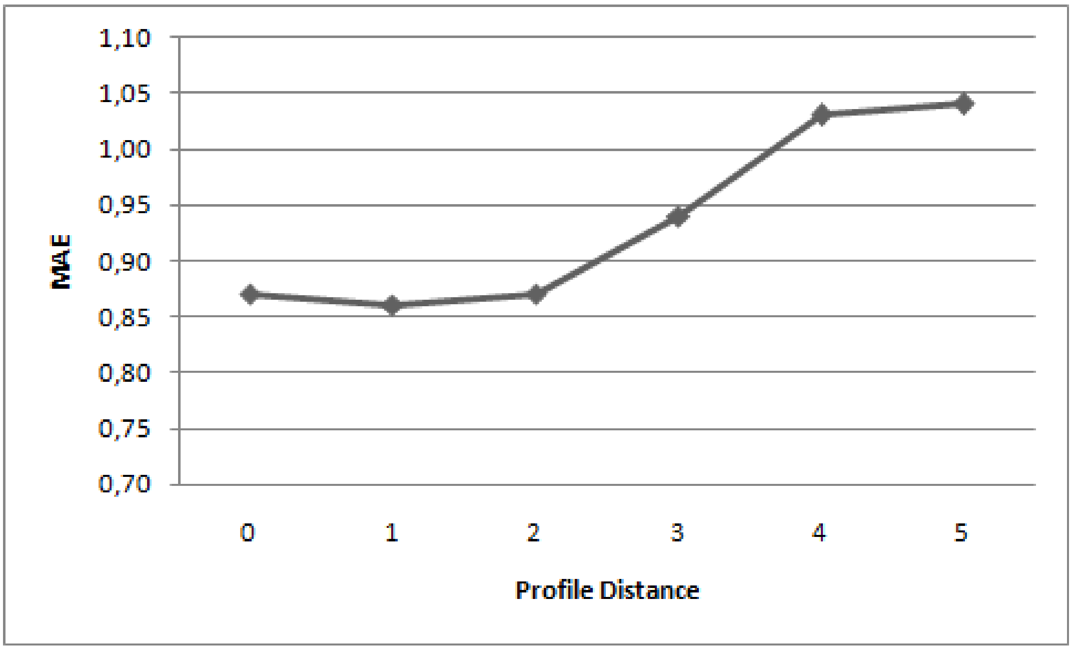


Figure 3: Plot of MAE for different profile distances.

theorem builds on a well know statistical principle, the bias-variance tradeoff, and its formulation is shown in Equation 6:

$$SqE(\bar{r}_{j,k,l}) = SqE(\hat{r}_{i,k,l}) - PDiv(\hat{r}_{i,k,l}) \qquad (6)$$

where $\bar{r}_{j,k,l}$ is the global predicted value for user j on profile k for item l. It is calculated as the arithmetic mean of the individual predictions, as it is shown in Equation 7:

$$\bar{r}_{j,k,l} = \frac{1}{K \times n_k} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \hat{r}_{i,k,l} \qquad (7)$$

where K is the total number or user profiles, n_k the number of users with profile k, and $\hat{r}_{i,k,l}$ the individual prediction of user i.

$SqE(\bar{r}_{j,k,l})$ is the squared error of the global prediction. It is calculated as shown in Equation 8, being $r_{j,k,l}$ the true value.

$$SqE(\bar{r}_{j,k,l}) = (\bar{r}_{j,k,l} - r_{j,k,l})^2 \qquad (8)$$

$SqE(\hat{r}_{i,k,l})$ is the squared error of the individual predictions used to compute the global prediction, the individual prediction being just the single rating generated by user i on profile k for item l. The error is calculated as shown in Equation 9:

$$SqE(\hat{r}_{i,k,l}) = \frac{1}{K \times n_k} \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\hat{r}_{i,k,l} - r_{j,k,l})^2 \qquad (9)$$

$PDiv(\hat{r}_{i,k,l})$ is the predictive diversity of the individual predictions used to compute the global prediction. It is calculated as shown in Equation 10

$$PDiv(\vec{s}) = \frac{1}{K \times n_k} \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\hat{r}_{i,k,l} - \bar{r}_{j,k,l})^2 \qquad (10)$$

The theorem states that the error of the collective prediction can be explained not only in terms of the error of the individual predictions, but also in terms of the diversity of the individual predictions. This is a possible explanation of the results shown in Tables 8 and 9. The aggregation of predictors that individually show higher MAEs could be balanced by the fact that the diversity of predictions increase in a higher rate. If this would happen in our case, the theorem could therefore explain our results.

6.1 Results

The application of the Diversity Prediction Theorem for predictors based on user profiles with increasing distances generated the results shown in Table 11. While the error values of $SqE(\bar{r}_{j,k,l})$ are different to MAE values of Table 7, a positive correlation with distances is also observed. However, the values of $SqE(\hat{r}_{i,k,l})$ and $PDiv(\hat{r}_{i,k,l})$ indicate that the reason behind the error increase is not because of the lower accuracy of the individual predictors at higher distances, but due to the poorer diversity of such predictors.

The unexpected results shown in tables 8 and 9 are explained

Table 11: Diversity Prediction Analysis for predictors based on user profiles with increasing distances: basic weighting prediction.

Distance	$SqE(\bar{r}_{j,k,l})$	$SqE(\hat{r}_{i,k,l})$	$PDiv(\hat{r}_{i,k,l})$
0	1.26678	2.14481	0.87803
1	1.19857	2.17998	0.98142
2	1.2202	2.19183	0.97163
3	1.48458	2.27204	0.78746
4	1.89438	2.24763	0.35325
5	1.85749	2.15699	0.2995

Table 12: Diversity Prediction Analysis for predictors based on aggregration of user profiles with increasing distances: basic weighting prediction.

Distance	$SqE(\bar{r}_{j,k,l})$	$SqE(\hat{r}_{i,k,l})$	$PDiv(\hat{r}_{i,k,l})$
0	1.26678	2.14481	0.87803
0+1	1.14378	2.15894	1.0139
0+1+2	1.10859	2.15455	1.04469
0+...+3	1.10515	2.16903	1.06132
0+...+4	1.10491	2.16971	1.0619
0+...+5	1.1046	2.16927	1.06234

in light of the application of the Diversity Prediction Theorem for predictors based on aggregation of user profiles. Table 12 show the results for increasing distances. It is clearly observed that error $SqE(\bar{r}_{j,k,l})$ decreases as the aggregation of user profiles increases with higher distances. This is explained by the values of $PDiv(\hat{r}_{i,k,l})$, which indicates that the diversity of predictors also increases with higher distances but at a higher rate than the errors of the individual predictors $SqE(\hat{r}_{i,k,l})$.

7. DISCUSSION

The first point to be discussed is the explanatory power of the Diversity Prediction Theorem. In order to test the aggregation effect in other algorithms, we applied the theorem to the traditional user-based algorithm. Table 13 illustrates how the error $SqE(\bar{r}_{j,k,l})$ again decreases as users with lower similarity to the target user are aggregated to the prediction equation. As in the case with our profile-based algorithms, this result can be explained by means of the increase of $PDiv(\hat{r}_{i,k,l})$.

The second point regards with our original hypothesis. The results show that profile-based algorithms only outperform traditional user-based algorithms when an aggregation scheme and a compensated weighting prediction are used. The aggregation works only when individual predictors with high prediction errors bring diversity to the pool of predictors. If this condition is satisfied, then the aggregation of such predictors could improve substantially the final prediction.

As a conclussion, profile-based algorithms can be particularly useful in the field of tourism in which users can show different profiles under different contexts. As future work we believe that preference learning, i.e. the discovery of user preferences associated to different user context, will be the key for improving predictions and generate better recommendations.

Table 13: Diversity Prediction Analysis for the User-Based Algorithm.

User Sim Th	$SqE(\bar{r}_{j,k,l})$	$SqE(\hat{r}_{i,k,l})$	$PDiv(\hat{r}_{i,k,l})$
1	1.19533	2.09803	0.9027
0.99	1.14034	2.08211	0.94185
0.98	1.11217	2.08478	0.972741
0.97	1.09727	2.08696	0.98983
0.96	1.09228	2.0948	1.00263
0.95	1.0924	2.10321	1.01078
0.94	1.09589	2.11771	1.02157
0.93	1.0983	2.12571	1.02696
0.92	1.09905	2.13274	1.03312
0.91	1.09871	2.13686	1.03751
0.90	1.10166	2.14358	1.04101
0.89	1.10268	2.14762	1.04392
0.88	1.10262	2.14945	1.04578
0.87	1.10197	2.1498	1.04681
0.86	1.10179	2.15093	1.04811
0.85	1.10062	2.15116	1.04958
0.84	1.10023	2.15226	1.0511
0.83	1.09975	2.15284	1.0522
0.82	1.10031	2.15422	1.0529
0.81	1.10057	2.15531	1.05368
0.80	1.10059	2.15545	1.05377
0.79	1.1007	2.15607	1.05425
0.78	1.10048	2.15584	1.05428
0.77	1.10067	2.15626	1.05446
0.76	1.1013	2.15743	1.05484
0.75	1.10089	2.15739	1.05531
0.74	1.10109	2.15806	1.05571
0.73	1.10103	2.15827	1.05598
0.72	1.10097	2.15818	1.05598
0.71	1.10078	2.15797	1.05602
0.70	1.10158	2.1592	1.05618
0.69	1.10138	2.15905	1.05629
0.68	1.10154	2.15928	1.0563
0.67	1.10168	2.15954	1.05635
0.66	1.10156	2.15957	1.05654
0.65	1.10158	2.15957	1.05652
0.64	1.10154	2.15958	1.05658
0.63	1.10165	2.15975	1.05659
0.62	1.1016	2.15955	1.05646
0.61	1.10154	2.15942	1.05641
0.60	1.10157	2.15972	1.05666

Acknowledgments

This research was sponsored by the Ministry of Science and Innovation of Spain under grant TIN2014-56633-C3-1-R as well as EMALCSA/Coruña Smart City under grant CSC-14-13.

8. REFERENCES

[1] Borrás, Joan, Antonio Moreno, and Aida Valls. Intelligent tourism recommender systems: A survey. Expert Systems with Applications 41.16, pages 7370-7389, 2014.

[2] Chihiro Ono, Mori Kurokawa, Yoichi Motomura, and Hideki Asoh. A Context-Aware Movie Preference Model Using a Bayesian Network for Recommendation and Promotion. *Proceedings of the 11th International Conference on User Modeling*, pages 257–267, 2007.

[3] Benedikt Engelbert, K Morisse, and KC Hamborg. Evaluation and user acceptance issues of a Bayesian classifier based TV Recommendation System. *Proc. of the RecSys 2012 Workshop on CARS*, 2012.

[4] Takashi Shiraki, C Ito, and T Ohno. Large Scale Evaluation of Multi-Mode Recommender System Using Predicted Contexts with Mobile Phone Users. *Proc. of the RecSys 2011 Workshop on CARS*, pages 4–8, 2011.

[5] Hernani Costa, Barbara Furtado, D Pires, Luis Macedo, and Amilcar Cardoso. Context and Intention-Awareness in POIs Recommender Systems. *Proc. of the RecSys 2012 Workshop on CARS*, pages 1–5, 2012.

[6] R. Vargas-Govea, B. and González-Serna, G. and Ponce-Medellín. Effects of relevant contextual features in the performance of a restaurant recommender system. *Proc. of the RecSys 2011 Workshop on CARS*, 2011.

[7] Breese, John S., David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Proc. of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.*, 1998.

[8] Lousame, Fabián P., and Eduardo Sánchez. A taxonomy of collaborative-based recommender systems. Web Personalization in Intelligent Environments. Springer Berlin Heidelberg, pages 81-117, 2009.

[9] John W. Tukey. Exploratory data analysis. Reading, Mass., 1977.

[10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.

[11] Lu Hong and SE Page. The Foundations of Collective Wisdom. *Collective Wisdom: Principles and Mechanisms*, pages 1–33, 2011.

TravelWithFriends: a Hybrid Group Recommender System for Travel Destinations

Toon De Pessemier
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
toon.depessemier@ugent.be

Jeroen Dhondt
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
jeroen.dhondt@ugent.be

Kris Vanhecke
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
kris.vanhecke@ugent.be

Luc Martens
iMinds-Ghent University
G. Crommenlaan 8 / 201
B-9050 Ghent, Belgium
luc1.martens@ugent.be

ABSTRACT

Recommender systems have proven their usefulness in many classical domains such as movies, books, and music to help users to overcome the information overload problem. But also in more challenging fields, such as tourism, recommender systems can act as a supporting tool for decision making when planning a trip. This paper proposes such a system providing group recommendations for travel destinations based on the users' rating profile, personal interests, and specific demands for their next destination. The proposed solution follows a hybrid approach, combining content-based, collaborative filtering, and knowledge-based strategies. Since traveling is often a group activity, families and groups of friends can receive group recommendations based on their combined profiles. The recommender system is tested in a prototype web application and evaluated by a group of test users. The results prove the usefulness of recommendations for travel destinations and show that the hybrid system outperforms each individual technique.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Recommender system, Hybrid, Travel, Tourism, Group recommendations

1. INTRODUCTION

Increasing amounts of information on traveling are available on the world wide web. As is the case for many other domains, the web is becoming the most important information source for planning a holiday. Specialized web sites, such as Expedia or SkyScanner, exist for finding the best deals, flight tickets or travel packages. Others, such as WikiVoyage or Frommers, are specialized in providing information and travel advice on different destinations. Reviews and evaluations of hotels, restaurants, and attractions can be read on websites such as TripAdvisor.

Although these services are all valuable information sources, they typically give no personal advice which holiday destination to chose. Here, recommender systems and artificial intelligence techniques [3] can help to overcome the problem of information overload and provide users valuable recommendations for destinations tailored to their personal preferences, requirements, and constraints.

Most research on recommender systems focuses on domains like movies, songs, or e-commerce. Specific characteristics of the domain make recommendations for travel destinations a lot harder. Firstly, data regarding travel destinations (metadata and ratings) are harder to acquire than the freely available dataset for movies such as MovieLens. Secondly, since most people travel only occasionally, the rating matrix is typically very sparse. Thirdly, users often have specific constraints (e.g., budget, distance) in addition to their personal preferences. And finally, traveling is typically a group activity: people often travel together. So group recommendations, combining the preferences of all group members, might be more suitable than individual recommendations.

The remainder of this paper is structured as follows. Section 2 gives an overview of related work. Section 3 provides an overview of the architecture of the travel recommender system and its internal data flow. Section 4 gives details about the data that is used and the data origins. In Section 5, the system is presented from the user point of view, with a focus on the features and the interface. The various recommendation algorithms are discussed in Section 6. Section 7 explains how the extension to group recommendations is realized. Section 8 gives the results of a user evaluation of the recommender systems. Finally, Section 9 draws conclusions from our research.

2. RELATED WORK

Various (group) recommender systems for points-of-interest (POIs), such as tourist attractions, restaurants, and hotels, have been proposed in literature. The Pocket Restaurant Finder provides restaurant recommendations for groups that are planning to go out eating together. The application can use the physical location of the kiosk or mobile device on which it is running, thereby taking into account the position of the people on top of their culinary preferences. Users have to specify their preferences regarding the cuisine type, restaurant amenities, price category, and ranges of travel time from their current location on a 5-point rating scale. When a group of people is gathered together, the Pocket Restaurant Finder pools these preferences together and presents a list of potential restaurants, sorted in order of expected desirability for the group using a content-based algorithm [14].

Intrigue is a group recommender system for tourist places which considers the characteristics of subgroups such as children or disabled and addresses the possibly conflicting preferences within the group. In this system, the preferences of these heterogeneous subgroups of people are managed and combined by using a group model in order to identify solutions satisfactory for the group as a whole [1].

Also in the context of tourist activities, the Travel Decision Forum is an interactive system that assists in the decision process of a group of users planning to take a vacation together [10]. The mediator of this system directs the interactions between the users thereby helping the members of the group to agree on a single set of criteria that are to be applied in the making of a decision. This recommender takes into account people's preferences regarding various characteristics such as the facilities that are available in the hotel room, the sightseeing attractions in the surrounding area, etc [9].

An alternative recommender system for planning a vacation is CATS (Collaborative Advisory Travel System) [15]. It allows a group of users to simultaneously collaborate on choosing a skiing holiday package which satisfies the group as a whole. This system has been developed around the DiamondTouch interactive tabletop, which makes it possible to develop a group recommender that can be physically shared between up to four users. Recommendations are based on the group profile, which is a combination of individual personal preferences.

The last example in the domain of POIs is Group Modeller, a group recommender that provides information about museums and exhibits for small groups of people [11]. This recommender system creates group models from a set of individual user models.

In contrast to existing systems, the goal of our recommender system, called TravelWithFriends, is to offer a more complete service delivering personalized recommendations for destinations taking into account the personal preferences, constraints, and feedback of the user. For each destination, travel distance, budget, and geographical location are considered and the local attractions and POIs are processed. Because of these domain specific characteristics, different recommender approaches are combined into a hybrid recommender. A group recommendation strategy is used to aggregate the preferences of different people who intend to travel together.

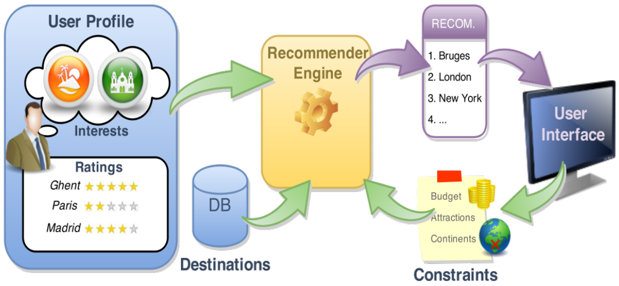


Figure 1: Overview of the data flow in the recommender system.

3. SYSTEM ARCHITECTURE AND DATA FLOW

Figure 1 shows the high-level flow of information through the recommender system. The recommender systems is fed with ratings and travel destinations coupled with metadata. Users interact with the system through the user interface. Personal constraints can be specified as input together with ratings for destinations. Recommendations are delivered as the output to the user, who can further give feedback on these recommendations.

Figure 2 zooms in on the recommender engine and the information flow within the recommender (red labels). The following subsequent steps can be indentified in the information flow.

1. Creating the user query: the user selects personal interests and destination constraints.

2. Constraint pre-filtering: the destinations in the database are checked against the constraints and a candidates shortlist is constructed.

3. Rating prediction: different recommendation algorithms calculate a rating prediction for the destinations of the shortlist.

4. Score merging: the rating predictions of the different algorithms are merged into one hybrid rating prediction.

5. Delivering recommendations: the destinations with the highest hybrid rating prediction are presented to the user as the final recommendations.

4. DATA STRUCTURE

The items, processed and output by the recommender, are all cities known for their tourism value. Many online services for POIs are available such as Google Places, Yelp, or Yahoo Local. Although these services contain lots of useful data, specific tourist information is often missing, such as information about tourist attractions in a city or the suitability of a location as a holiday destination.

As information source for our recommender service, we used the freely available data set of WikiVoyage [19], the Wikipedia alternative for travel destinations, which is available under an open license by the Wikimedia Foundation. This information service consists of more than 26,000 locations and tourist information pages, created by users. One

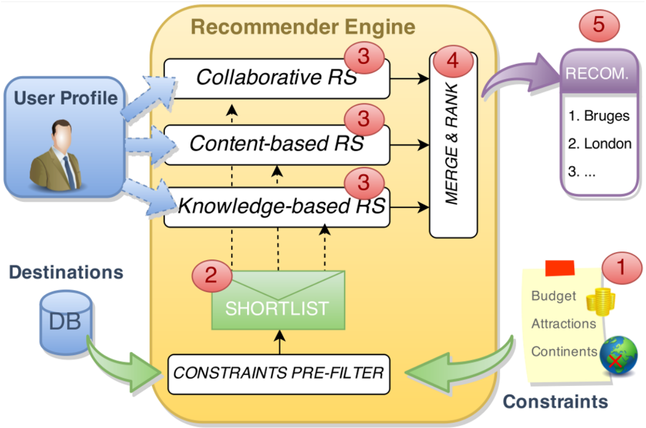


Figure 2: The system architecture of the recommender engine.

of the main advantages of this service, is that all entries have specific tourism value and come with information that is useful for tourists. However, many of these pages are not actual destinations but rather collections of destinations, information on a specific tour, etc. Therefore, a first filtering of the entries of WikiVoyage was performed using the database of GeoNames [7] in order to select only the actual destinations. GeoNames is a database listing over 100,000 place names in the world with their geographic data. The result of this first filtering was a set of 6,900 cities, towns, and villages.

Many of the resulting listings are minor, little-known locations, which may be interesting to explore while in the vicinity, but that have insufficient tourism value to be a travel destination on itself. Since these minor locations would be unsuitable as a recommendation for a travel destination, a second filter was necessary in order to only recommend 'sufficiently relevant' places. This filter used the popularity (measured by the number of ratings) on the popular website TripAdvisor [18], an American travel website providing reviews of travel-related content. The threshold for being considered as sufficiently relevant for a tourist destination was set to having at least 25,000 reviews on TripAdvisor. The resulting database contains 685 famous (and less famous) tourist locations, but can be easily extended with additional destinations (by relaxing one of the filters for example).

Regarding the information about the travel destinations, two crucial information resources are consulted:

- The *Travel Destination* database consists of general information about the destination, such as a description and location coordinates, as well as background information on the region and country.
- The *Domain Knowledge* database consists of specific domain knowledge such as a mapping of locations and typical tourist profiles, attraction types, and typical transport costs.

In order to obtain a typical tourist profile for each destination, the website Gogobot [8] is consulted. Gogobot is a travel application website that lets users rate travel destinations and attractions. In comparison with other social travel networks such as TripAdvisor, Gogobot differentiates by making use of *tribes*. Gogobot's 19 tribes represent tourist profiles (e.g., backpackers, family travelers, adventure travelers, business travelers, or budget travelers) to which users may relate. The tribe-specific information for a destination is obtained in two ways. On the one hand, users on Gogobot can explicitly specify that a destination is 'recommended for' a specific tribe such as Backpackers. On the other hand, Gogobot users can indicate in their own profile which tribes best match their interests. Destinations that received a positive rating from the user may also be suitable to other users who belong to (some of) the same tribes. In other words, we assume an implicit coupling between the user's tribes and the destinations that the user has rated. By gathering the tribe information of all users who rated the destination, a more detailed profile of the destination can be obtained. When combining tribe information of different users, the explicit tribe association was given twice the weight of the implicit association. In case of a user-item pair for which an explicit tribe recommendation as well as an implicit tribe association based on a star rating is available, only the explicit tribe recommendation is used.

For travel costs, a specialized information service was used. Various web services provide real-time prices for trains, airplanes, or another means of transportation. The webservice Rome2rio [17], which was used in TravelWithFriends, combines different transportation methods and predicts the travel cost between any two locations in the world. It taps into the information of many different online services and databases to gather information on flights, trains, buses, boats, and even taxi fares to come up with all possible means to reach your destination.

The users, who interact with the system and receive recommendations, are also represented by two information resources:

- The *User Rating* database keeps track of the 5-star ratings of all users given to travel destinations, as well as implicit feedback that indicates which places the user has visited (without star rating).
- The *User Profile* database stores more general information about each user such as login information, explicitly stated interests, and demographic data.

To reduce cold-start difficulties of our recommender system, ratings and implicit feedback (selecting "Been here" to indicate that you have visited the location) from Gogobot were used. More than 300,000 ratings by 1759 users from Gogobot were imported. Ratings for attractions were aggregated to ratings for the destination were the attraction can be found. Ratings for destinations that are not in the destination database (or filtered out because of their low tourism value), are redundant and ignored in the calculations. Finally, 53,028 ratings were imported into the recommender system.

5. TRAVELWITHFRIENDS WEB APPLICATION

The TravelWithFriends recommender system is made available for end-users through a web application accessible in a standard web browser. The web application consists of many pages, such as the register page, a page for creating and joining groups, and the traditional search functionality.

In comparison with other recommender domains, traveling is for many users a less frequent activity compared to

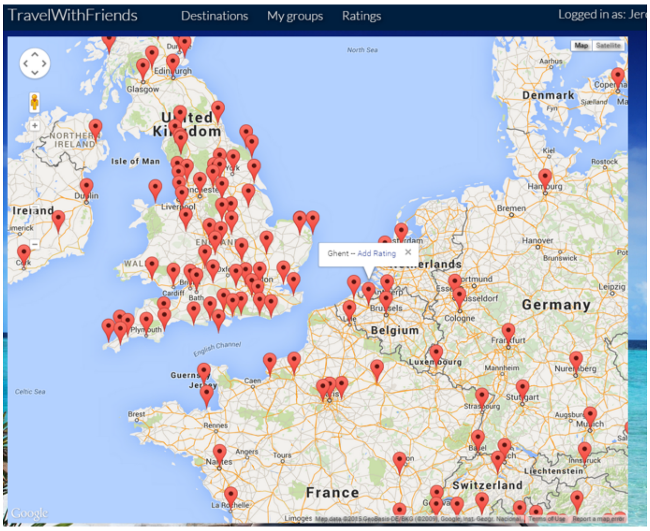


Figure 3: Screenshot of the Google Maps based user interface for destination selection.

Figure 4: Screenshot of the user interface showing the possibility to explicitly specify preferences.

listening to songs or watching movies, thereby exacerbating the sparsity problem. To reduce the sparsity, users can state their previous travel experiences by giving ratings to destinations they have visited in the past. Users can search for these locations by name, or alternatively, they can navigate to the location through Google Maps, as demonstrated in Figure 3.

To bootstrap the content-based recommender component, users can also indicate their interests for 19 travel categories, which are used as an initial profile. Figure 4 shows a screenshot of the user interface illustrating the explicit profile preferences of the user. For each of the typical travel interests, users can specify their affinity.

In addition, users can specify personal constraints regarding the travel destination, such as budget, the continent of the destination, and the presence of specific kinds of attractions.

6. RECOMMENDER ENGINE

To cope with the complex aspects of travel recommendations, such as the desired serendipity, the sparsity problem, and user constraints, multiple recommender approaches are combined into a weighted hybrid recommender. Collaborative filtering can introduce serendipity into the recommendations by comparing consumption data of similar users. The content-based approach can better handle the sparse data matrix. User constraints are taken into account by the knowledge-based recommender (and a pre-filter).

In a production environment, users will receive only one list of (hybrid) recommendations. However, for evaluation purposes, the test users (cfr. Section 8) received five recommendations lists so that each algorithm could be evaluated separately: collaborative filtering, content-based, knowledge-based, hybrid, and a static list of the most popular travel destinations. Users can select a destination from the recommendation list to request more information, or can give feedback on the recommendations.

Before applying the recommendation algorithms, a pre-filtering of the candidate destinations is performed. The system eliminates destinations that the user has already visited (assuming these would make for undesirable recommendations) as well as destinations that do not fulfill the user's constraints. In the next phase, the various recommendation algorithms will restrict their selection to destinations that made the shortlist.

6.1 Collaborative filtering

For collaborating filtering, we opted for an item-item approach and used the implementation of the Lenskit Framework [6]. As input, a combination of explicit ratings and implicit feedback is used, given by users on Gogobot.com [8]. For the explicit ratings, the system uses ratings for the destination (the city), as well as ratings for the attractions at the destination which are averaged into one rating. In comparison with other domains, such as online shop items or music, the ratings gathered from Gogobot are much more positive (more than 90% is ≥ 3).

In addition, for each destination or attraction at a destination, users can indicate on Gogobot if they have been there. This data is used as implicit feedback on the destination in TravelWithFriends, because it contains two pieces of information: 1. the user has been to this place, and recommending this destination again is therefore undesirable and 2. the user has shown interest in this place by visiting it.

The item-based collaborative approach takes two phases to predict the rating of a user for a given destination. In the first phase, a collection of most-similar destinations, called the k-nearest neighbors, is determined. These neighbors are selected by calculating a similarity measure between each pair of destinations and selecting the ones with the highest similarity value. For the neighbor selection, explicit ratings combined with implicit feedback were used in order to reduce the sparsity of the matrix. In our approach, explicit and implicit feedback are mapped to a binary value: 1 if the destination was rated or tagged as "Been here"; 0 otherwise. Since ratings are mainly positive, the mapping of 5-star ratings to a binary value is not considered as a loss of information for calculating item similarities. Because of these binary values, the traditional Pearson correlation is not feasible. Because some items (popular destinations) received much more feedback than others, the cosine similarity is not the optimal correlation measure to ensure that less popular destinations have a fair chance to get recommended. Another approach is to use the conditional probability $P(j|i) = \frac{P(i,j)}{P(i)}$ to calculate how likely a specific destination is, in case another destination has been visited. This function evaluates whether two items are associated, but still favors destinations with a large amount of feedback, i.e. highly popular destinations.

To correct for the popularity of the item j, a modified version of the conditional probability with an additional term $P(j)^\alpha$ was used as similarity measure:

$$sim(i,j) = \frac{P(i,j)}{P(i) * P(j)^\alpha} \quad (1)$$

In the implementation, $\alpha = 0.2$ provided the best balance between constraining the popularity and measuring similarity based on empirical research.

In the second phase, the rating prediction will be calculated based on these most similar destinations. In our implementation, $k = 20$ was chosen as this is a typical value for the k-nearest neighbors algorithm. We denote $\mathcal{N}_u^e(i)$ as the neighborhood of destination i for user u. This neighborhood consists of the items j that the target user u has explicitly rated, and that are most similar to the item i. As such, this neighborhood is different for each user. Next, the weighted sum scoring function with mean centering [5] is used for items that received an explicit rating from the target user in order to make a rating prediction $\hat{r}^e$.

$$\hat{r}^e_{u,i} = \overline{r}_i + \frac{\sum\limits_{j \in \mathcal{N}_u^e(i)} sim(i,j)(r_{u,j} - \overline{r}_j)}{\sum\limits_{j \in \mathcal{N}_u^e(i)} |sim(i,j)|} \quad (2)$$

To also take into account the implicit feedback, a second scoring function was used for the binary data.

$$\hat{r}^i_{u,i} = \frac{\sum\limits_{j \in \mathcal{N}_u^i(i)} sim(i,j) * \overline{r}_j}{\sum\limits_{j \in \mathcal{N}_u^i(i)} |sim(i,j)|} \quad (3)$$

Here, the neighborhood $\mathcal{N}_u^i(i)$ stands for the items j for which the target user has provided implicit feedback or an explicit rating, and that are most similar to the item i. Notice that the user's neighborhood for implicit feedback $\mathcal{N}_u^i(i)$ can be different from the user's neighborhood for explicit feedback $\mathcal{N}_u^e(i)$, since a user might have provided implicit feedback for different items compared to the user's ratings.

Finally, the weighted sum of both rating predictions is calculated to combine explicit ratings and implicit feedback, as is commonly done [20].

$$\hat{r}_{u,i} = \frac{\alpha * \hat{r}^e_{u,i} + \beta * \hat{r}^i_{u,i}}{\alpha + \beta} \quad (4)$$

The weights α and β were set to: $\alpha = 2 * \#\mathcal{N}_u^e(i)$ and $\beta = \#\mathcal{N}_u^i(i)$. The values of α and β were chosen so that if both neighborhoods contain the same number of items (i.e. 20 if enough neighbors can be found), then the rating prediction for the explicit ratings contributes for 2/3 versus only 1/3 for the prediction based on implicit feedback. If however, the neighborhood for the explicit ratings has far fewer similar items than the one for implicit feedback, then the weight is shifted more towards the rating prediction with the implicit feedback.

If data sparsity prevents finding an extensive neighborhood, and $\mathcal{N}_u^i(i)$ contains fewer than 5 similar items (which implies $\mathcal{N}_u^e(i)$ has also less than 5 items), then the collaborative filtering approach is considered unreliable. In this case, recommendations using collaborative filtering might not be accurate enough given the small neighborhood size and the recommendations are disregarded. The recommender system will then fall back on the content-based and knowledge-based approaches.

6.2 Content-based recommender

The idea of content-based recommendation approaches is to find matches between features of a particular item and the user's profile. If item features are not directly available, they are often obtained by analyzing textual descriptions of the items and extracting keywords from them. This approach can also be applied in the domain of travel destinations, but has been shown to deliver often irrelevant or overly obvious features. Therefore, in TravelWithFriends another approach, specially tailored for the domain of travel destinations, was adopted.

The approach is based on the idea to characterize a travel destination by the categories and keywords linked to the POIs at the destination. These POIs are often accurately annotated by specialized information services and are often the main incentive to visit a travel destination. TravelWithFriends utilizes the tags of attractions described on TripAdvisor [18], but similar information sources can be a valuable alternative. The tags of attractions on TripAdvisor are chosen from a fixed set of attraction categories and are restricted to one tourism topic. We illustrate our approach using Paris as a potential destination. Among its most prominent tourist attractions are the world famed museums 'musée du Louvre' (categorized as [Art Museum, Museums] on TripAdvisor), and 'musée d'Orsay' [Speciality Museum, Museums]. Paris features also some well known landmarks such as the 'Eiffel tower' [Points of Interest & Landmarks, Sights & Landmarks], 'Arc de triomphe' [Architectural Buildings, Historic Sites, Sights & Landmarks] and the 'Notre Dame Cathedral' [Religious Sites, Sights & Landmarks]. These key attractions and their associated tags already give a good overview of what Paris has to offer to tourists.

The relative importance of a tag for an item is typically determined by a measure such as the TF-IDF (Term Frequency - Inverse Document Frequency) [12]. To increase the contribution of the more famous and popular attractions at the destination, the tag frequency is multiplied by the number of reviews for the coupled attraction. In the example of Paris, the tag 'Speciality Museums' (attached to musée d'Orsay) was applied 26,149 times (the number of reviews for musée d'Orsay) to Paris. In contrast, the tags applied to the Parc des Buttes Chaumont (i.e. the 50th most popular attraction in Paris) only receives a weight of 548, the number of reviews for Parc des Buttes Chaumont.

Because of a large variation in the frequency of occurrence of tags and reviews, a minor change was made to the traditional TF-IDF by taking the square root of the frequency term, $f_{t,d}$ to reduce the influence of the absolute review frequency. This results in the following formula for the TF-IDF weight for tag t of destination d, part of the collection of all destinations D. Here, N is the number of destinations in D, and $f_{t,d}$ is the frequency of tag t in destination d, which means the frequency of tag t in all attraction descriptions of destination d, multiplied by the number of reviews for that attraction.

$$TFIDF(t,d,D) = \sqrt{f_{t,d}} * log_2 \frac{N}{|\{d \in D : t \in d\}|} \quad (5)$$

The necessity to take the square root of the term frequency can be illustrated by an example. If the tag frequency, multiplied by the number of reviews, is used in combination with the traditional TF-IDF, then the weight of a few top attractions is too high, thereby neglecting the contribution of other attractions at the destination. If any of these top attractions has a rare tag (and thus a very high IDF), this tag will dominate the recommendations. For 'Barcelona' as destination, for instance, the 'Sagrada Familia' is one of the top attractions, which has a rather rare tag 'Religious Sites'. This tag will dominate the recommendations in case of the traditional TF-IDF, leading to "similar" destinations, all renown for their beautiful cathedrals including 'Santiago de Compostela', 'Cologne', and 'Rouen'. Since Barcelona offers much more than the 'Sagrada Familia', this biased reflection was undesirable.

The logarithm of the term frequency has been proposed as an alternative weight for the term frequency in literature [12]. However, experiments showed that the logarithm shifted too much weight to less popular attractions. Analysis of the resulting recommendations showed that for the domain of travel destinations, the square root of the term frequency provides the right balance between both popular and less popular attractions. The square root reduces the weight of top attractions, but preserves a sufficiently large difference in contribution compared to less significant attractions.

In the same manner, the derived destination tags are used to build a content-based user profile based on the destinations that are positively rated (≥ 3.5) by the user. For all these positively rated destinations, the TF-IDF values are summed per tag in the user profile. Finally, the derived destination tags are compared with the user profile using the traditional cosine similarity. The resulting similarity score is transformed to the range $[1-5]$ and used as content-based rating prediction.

6.3 Knowledge-based recommender

The knowledge-based recommenders makes use of deeper connections and information provided by domain experts. Just like a human travel agent typically asks customers for their target budget, travel distance and accommodation expectations, the knowledge-based recommender will select destinations in a similar matter. This user input can be defined as a hard constraint or as a soft constraint (rather a guideline for the recommender). The pre-filter eliminates all destinations that do not fulfill the hard constraints before the candidate destinations are handed over to the recommendation algorithms. The soft constraints are handled by the knowledge-based recommender, which gives a penalty to destinations that are a good match for the user's preferences, but do not completely fulfill the requirements. So, destinations for which the soft constraints are not met, can still end up in the final recommendation list if they match the user's preferences.

In comparison with collaborative filtering and the content-based recommender, the knowledge-based recommender collects information specific for the domain of travel destinations, and is therefore not directly applicable to other domains. The following information sources were integrated into the knowledge based recommender:

1. Geographic information: the exact location (longitude and latitude), continent, and country of each destination.
2. Travel costs: the costs of traveling from your current location to the destination in question.
3. Attraction types: what specific attraction types can be found at that destination.
4. Tourist profile (stereotypes such as Backpackers, Family Travelers, etc.): to what degree each location matches typical tourist profiles as defined in Gogobot [8].

Constraints regarding the location and distance, as well as the traveling cost can be specified by the user in the interface of the application, as showed in Figure 5. Requirements regarding the types of attractions available at the destination, such as beaches, amusement parks, etc., can be selected using check-boxes. These constraints and user requirements are matched against the candidate destinations, providing a score for each dimension (location, costs, profile, attractions).

Table 1 shows the scoring function for each dimension, as well as a weight for the relative contribution of each dimension to the rating prediction. For the location dimension, a score function is proposed that decreases as the travel distance exceeds the *max_distance* as defined by the user. The square root allows destinations that are only slightly further than the *max_distance*, by assigning only a small penalty to these. For the cost dimension, a score function is proposed that decreases linearly as soon as the expected cost exceeds the predefined budget of the user. For the attractions available at the destination, the scoring is the ratio of the number of attractions that are requested and available, and the total number of attractions that are requested by the user.

For the tourist profile, each user is linked to one or more typical profiles (e.g., 30% Backpackers, 70% Adventure Travelers). This mapping to typical profiles can be performed in two ways. Users have the option to manually select what profiles they believe best match their interests (Figure 4). Alternatively, the typical profiles can be selected automatically by matching the user's explicit ratings with the typical profiles of the rated destinations. This approach is similar to the profile creation based on tags, used in the content-based recommender.

For each candidate destination a typical tourist profile is calculated based on the typical profiles of the users who rated the destination on Gogobot [8]. E.g., if 80% of the users who positively rated the destination are Backpackers, then it is classified as a 80% Backpackers destination. Subsequently, the user's typical profile is compared with the destination's typical profile using the cosine similarity. The scores of the different dimensions are combined using a weighted average to calculate the knowledge-based rating prediction.

$$\hat{r}_{u,i} = \frac{\sum_{k \in \mathcal{D}} w_k * sc(i,k)}{\sum_{k \in \mathcal{D}} w_k} \tag{6}$$

Here, the summation is limited to the dimensions D, for which constraints are specified by the user. The weights of the different dimensions are specified in Table 1. In our implementation, all weights have the same value, except for the

Table 1: The scoring function of the components of the knowledge-based recommender

Dimension	Scoring sc(i,k)	w_k
Geo location	$1 - \sqrt{\frac{max(distance(user, location) - max\_distance,\ 0)}{max\_distance}}$	1
Travel costs	$1 - \frac{max(expected\_cost - max\_budget, 0)}{max\_budget}$	1
Attractions	$\frac{\#typesmatched}{total\#typesrequested}$	$\frac{1}{2}$
Tourist profile	cosine sim(item, profile)	1

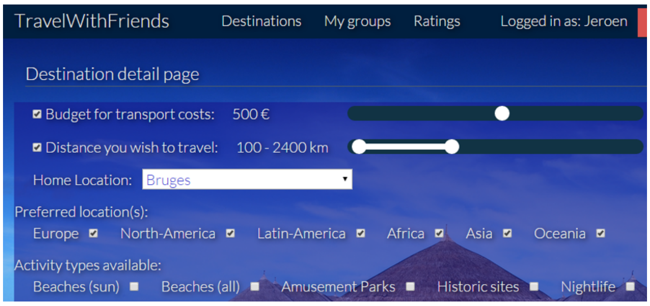


Figure 5: Screenshot of the user interface showing the options to define user constraints.

weight of the attractions. Since users might specify multiple attraction types that are sometimes hard to combine (e.g., beaches, amusement parks, and historic sites), the weight of the attraction dimension was decreased to 1/2.

6.4 Hybrid recommender

While the three individual recommendation approaches each generate a rating prediction, merging their output combines the different information sources and should make up for misjudgements of the individual recommenders. To merge the rating predictions, a simple weighted sum of all three predicted scores is calculated. The different indices are cf for collaborative filtering, cbf for content-based filtering, and kb for the knowledge-based recommender.

$$\hat{r}_{hybrid} = w_{cf} * \hat{r}_{cf} + w_{cbf} * \hat{r}_{cbf} + w_{kb} * \hat{r}_{kb} \tag{7}$$

$$w_{cf} + w_{cbf} + w_{kb} = 1 \tag{8}$$

These weights are not static, but influenced by the available data. If enough data is available for all recommenders, each algorithm will contribute for 1/3 to the hybrid recommendations. If only a limited amount of neighbors are found for collaborative filtering, w_{cf} is lower, or even zero if less than 5 neighbors can be found. The knowledge-based recommender has a lower contribution ($w_{kb} < 1/3$) if fewer (soft) constraints are specified by the user. Since an initial profile is created for each user, the content-based recommender usually has sufficient information to generate recommendations and can therefore act as the fall-back algorithm when both other approaches show little confidence.

7. GROUP RECOMMENDATION

Many travel plans are not made by individuals but by groups of people: friends, families, sports teams, etc. Besides individual recommendations, TravelWithFriends therefore allows users to create groups or join existing groups of friends and receive recommendations for the whole group. For most people, choosing a travel destination is an important decision, in which communication among the group members is essential. Group members typically want to discuss the destination thereby communicating their concerns and preferences based on some concrete suggestions. Therefore, group recommendations are generated in two subsequent phases. First, the system makes a shortlist of destinations for the group, based on the recommendation lists of each individual group member. Second, the group recommender acts as a conversational recommender. Each group member has the opportunity to provide feedback and rank this shortlist of candidates, after which the system makes a fair and balanced review and presents the final recommendations.

The process of generating group recommendations is illustrated in Figure 6. In the first phase, recommendations of individual users are merged into group recommendations using a recommendation aggregation technique [4]. We opted to aggregate the individual recommendation lists into a group recommendation list instead of aggregating the individuals' ratings into group ratings and subsequently generating group recommendations from these group ratings [4]. The reason for this is that aggregated recommendations for a group can be linked to recommendations for individuals, and as a result, can be explained more clearly in terms of an individual's preferences and constraints.

To aggregate the individual recommendations into group recommendations, various strategies are possible [13]. However, some have obvious disadvantages. Using the average of each member's rating prediction as a rating prediction for the group, i.e. the 'average' strategy, has the disadvantage of individuals who might be very unhappy with the final choice. If one user has a strong aversion to a particular destination, but the other group members love it, then this destination might still be recommended because of a high average rating prediction. Leaving one of the group members really unhappy about the destination is an unwanted situation. Therefore, the 'average without misery' strategy is employed as recommendation aggregation method, since this strategy cares about fairness and avoiding individual misery [13]. This strategy calculates the average of each member's rating prediction, but eliminates the destination if one of the group members has a rating prediction below a threshold. The threshold was chosen at 50% of highest scoring destination for that user. This way, destinations that are strongly disliked by any of the group members are eliminated from the group recommendations. Based on the assumption that users want recommendations for destinations they have not yet visited, destinations that have already received feedback or a rating from one of the group members are also eliminated. The result is a list of ten candidate destinations which are offered as an initial recommendation list

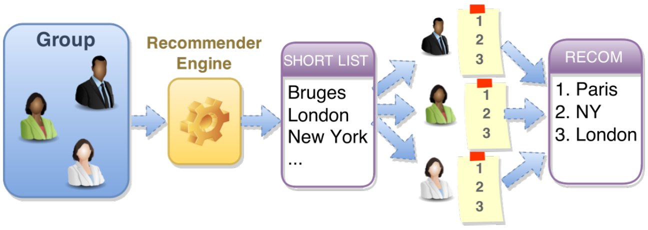


Figure 6: Schematic overview of the generation of group recommendations.

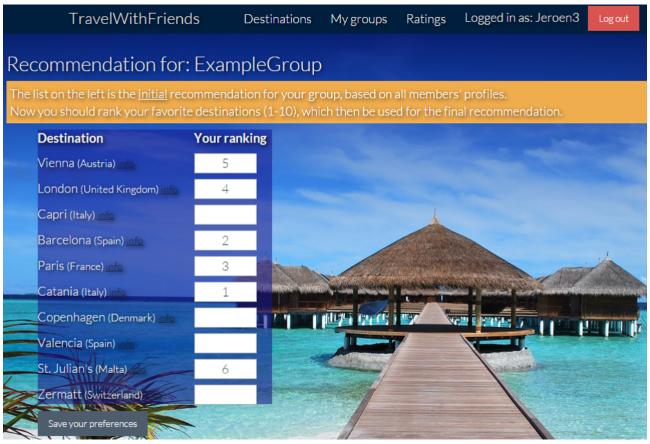


Figure 7: Screenshot of the user interface showing the possibility to rank the group recommendations.

to the group.

In the second phase, group members can give feedback on the list and indicate their favorites. In order to give users the opportunity to negotiate the travel options, each member is invited to give a personal ranking (from 1 to 10) to the candidate group recommendations, as shown in Figure 7. The users' ranking of the recommendations are processed by the Borda count election method. The Borda count method determines the winner(s) of the election by giving each candidate a number of points corresponding to the number of candidates ranked lower [13]. Based on the resulting Borda count, the group is finally presented their top-5 destinations to reduce the choice overload of the final recommendation list.

8. EVALUATION

The TravelWithFriends application was presented to 16 users who were asked to experiment with the recommender system and evaluate the different recommendation lists (3 algorithms, hybrid recommendations, and a static list of popular destinations). The evaluation was performed in two phases.

The goal of the first phase was to assess the general quality of the travel recommender system. To collect some qualitative feedback regarding the service, each user was asked to fill in a questionnaire, based on the evaluation framework of Pu [16]. Figure 8 shows the results of four multiple-choice questions assessing the general quality of the system (not about a specific recommendation algorithm). All users were overall satisfied with the system (Figure 8(a)). Their comments (not shown here) were positive about the possibility to explore new, unfamiliar destinations. They enjoyed the experience of determining their next travel destination using the service.

Next, the results show that most users consider it easy enough to specify their preferences. However, there is some room for improvement here (Figure 8(b)). The open questions indicated users would like more options for choosing their type of holiday (citytrip vs. hiking trip), the option of a general safety advice, the tourist-friendliness of the destination, and the option to determine the duration of the trip.

In addition, most users are convinced that the recommendations are useful and a suitable candidate for their travel destination (Figure 8(c)). Adding explanations to the recommendations can be an improvement to further increase the users' trust in the system. Finally, almost all users also indicated they would use the application if it became publicly available (Figure 8(d)).

The goal of the second phase was to assess the users' opinion about the quality of each recommendation algorithm: collaborative filtering (CF), content-based filtering (CBF), the knowledge-based recommender (KB) and the hybrid combination of these algorithms (HYB). As a baseline to compare the different algorithms, a fifth approach was included, which simply returned the static, non-personalized top of most-popular destinations (TOP). This list shows the most rated destinations on TripAdvisor, excluding the destinations already rated by the user.

Users were invited to use the application, starting with the preparatory steps of adding some ratings, selecting interests, and specifying constraints. Subsequently, users could explore the recommendations generated based on their input. To compare the different recommendation algorithms, users in this test were presented with five different lists of eight recommendations each. Eight recommendations is considered as an optimal number to prevent choice overload, while providing users different options and the coupled choice satisfaction [2]. These five lists were randomly shuffled and presented without any hint of the algorithm that was used to produce the list in order to obtain unbiased evaluation results.

The test users were asked to rank these five lists based on their own assessment of the most suitable recommendations. Figure 9 shows the distribution of the obtained rankings for each algorithm. These results indicate that the hybrid algorithm is most appreciated by the test users with 6 users choosing this as the best option, and 5 more users rewarding this algorithm with a second place.

Besides the hybrid recommender, also the content-based and knowledge-based recommender were liked by many users, whereas the TOP approach achieved the worst results (as expected). We hypothesize that content-based and knowledge-based recommendations score better than the collaborative filter because users recognize their constraints and personal preferences in these recommendations.

A statistical analysis using the Student's t-test was performed to test the superiority of the recommendation algorithms against the baseline approach (TOP). The mean of the rankings assigned by the users was compared for the different algorithms. The null-hypothesis was that the differences in mean ranking were merely due to randomness of

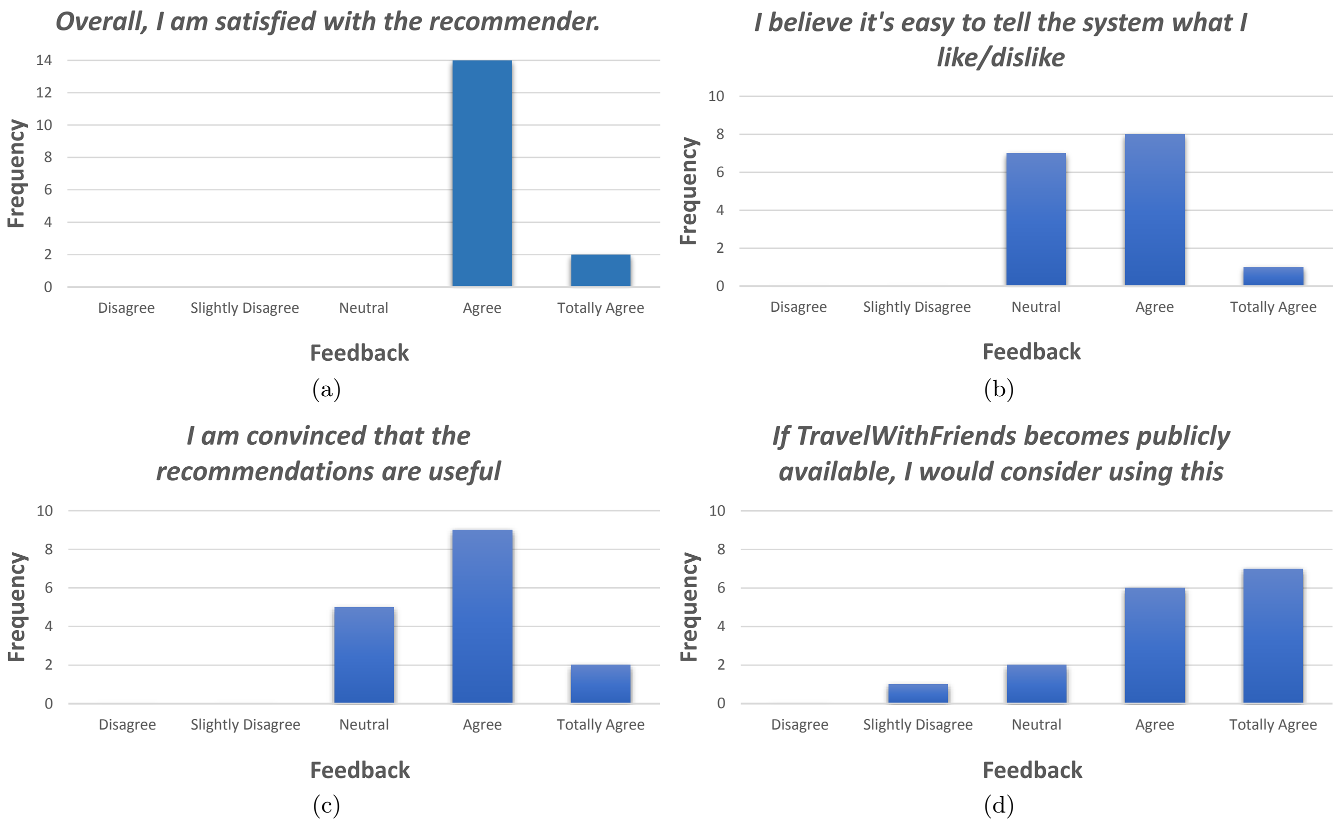


Figure 8: The results of the user evaluation regarding the general quality of the travel recommender.

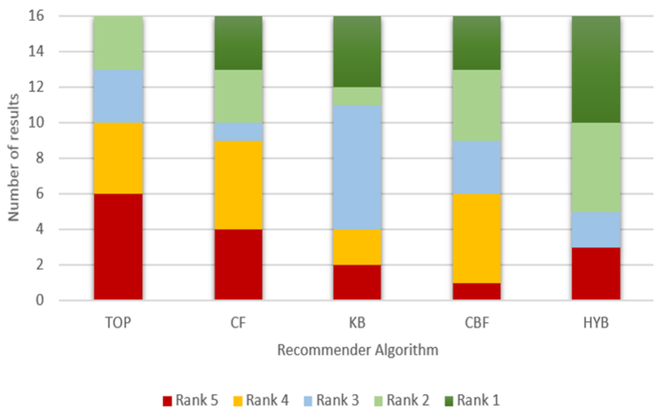


Figure 9: Distribution of the rankings given to the recommendation algorithms by the test users. Rank 1 is the best; Rank 5 is the worst.

the results. The t-tests showed that the difference with the baseline recommender (TOP) in terms of mean ranking was statistically significant for the hybrid recommender (p-value = 0.004), the content-based recommender (p-value = 0.028), and the knowledge-based approach (p-value = 0.031). Only the collaborative filter did not show statistical evidence (p-value = 0.251) of receiving a better ranking than the baseline.

9. CONCLUSIONS

Because travel destinations proved to be a complex domain for recommendations, characterized by personal preferences, user constraints, and being a group activity, no single algorithm would be able to consider all aspects. Moreover, gathering metadata and user feedback (ratings) showed to be less trivial for travel destinations than for more classical recommender domains such as movies or books. A hybrid system, combining different recommender approaches supplemented with the ability to generate group recommendations, was proposed.

User testing showed the usefulness of the proposed travel recommender system. Users enjoyed the new approach for discovering destinations and were happy to explore new places to consider as a travel destination. A comparison of different recommendation algorithms indicated that users prefer the hybrid recommendations above content-based, knowledge-based, and collaborative filtering recommendations. Differences in recommendation quality between these algorithms and an unpersonalized list of the most-popular destinations are clearly noticeable to the users. User comments argued for the inclusion of explanations of the recommendations in future versions of the application. Another option for future work is to recommend close-by locations, a multi-destination holiday, or a wider region to explore. In addition to the evaluation by individual users, the system will be tested by groups of users to evaluated the two phase group recommendation process in the future. Given the impact of the knowledge-based approach, it will be considered for pre-filtering plus weight initialization of the destination candidate set, rather than a recommender itself. Also the combination of both knowledge-based and content-based techniques will be investigated, because collaborative filtering seems to decrease the satisfaction of the users. Finally, we plan a performance evaluation to make the system useful as an actual product.

10. ACKNOWLEDGMENTS

We would like to thank Jeroen Dhondt for the work he performed in the context of this research during his master thesis.

11. REFERENCES

[1] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Tailoring the recommendation of tourist information to heterogeneous user groups. In S. Reich, M. Tzagarakis, and P. De Bra, editors, *Hypermedia: Openness, Structural Awareness, and Adaptivity*, volume 2266 of *Lecture Notes in Computer Science*, pages 228–231. Springer Berlin Heidelberg, 2002.

[2] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus. Understanding choice overload in recommender systems. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 63–70, New York, NY, USA, 2010. ACM.

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

[4] T. De Pessemier, S. Dooms, and L. Martens. Comparison of group recommendation algorithms. *Multimedia Tools and Applications*, 72(3):2497–2541, 2014.

[5] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, 2011.

[6] M. D. Ekstrand, M. Ludwig, J. Kolb, and J. T. Riedl. Lenskit: A modular recommender framework. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 349–350, New York, NY, USA, 2011. ACM.

[7] GeoNames. The geonames geographical database, 2015. Available at `http://www.geonames.org/`.

[8] Gogobot Inc. The go-to place for places to go, 2015. Available at `http://Gogobot.com/`.

[9] A. Jameson. More than the sum of its members: challenges for group recommender systems. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 48–54, New York, NY, USA, 2004. ACM.

[10] A. Jameson, S. Baldes, and T. Kleinbauer. Two methods for enhancing mutual awareness in a group recommender system. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 447–449, New York, NY, USA, 2004. ACM.

[11] J. Kay and W. Niu. Adapting information delivery to groups of people. In *Proceedings of the Workshop on New Technologies for Personalized Information Access at the Tenth International Conference on User Modeling*, 2006.

[12] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[13] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. In *Personalized Digital Television*, volume 6 of *Human-Computer Interaction Series*, pages 93–141. Springer Netherlands, 2004.

[14] J. McCarthy. Pocket restaurantfinder: A situated recommender system for groups. In *Proceedings of the Workshop on Mobile AdHoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*. ACM, 2002.

[15] K. McCarthy, M. Salamo, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Cats: A synchronous approach to collaborative group recommendation. In G. Sutcliffe and R. Goebel, editors, *FLAIRS Conference*, pages 86–91. AAAI Press, 2006.

[16] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 157–164, New York, NY, USA, 2011. ACM.

[17] Rome2rio. Rome2rio api and whitelabel, 2015. Available at `http://www.rome2rio.com/documentation`.

[18] TripAdvisor Inc. Tripadvisor website, 2015. Available at `http://www.tripadvisor.com/`.

[19] Wikivoyage Inc. Wikivoyage the free worldwide travel guide that anyone can edit, 2015. Available at `http://dumps.wikimedia.org/enwikivoyage/`.

[20] M. Zanker and M. Jessenitschnig. Collaborative feature-combination recommender exploiting explicit and implicit user feedback. In *Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on*, pages 49–56, July 2009.

CLG-REJA: A Consensus Location-aware group recommender system for Restaurants

Jorge Castro
Department of Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
jcastro@decsai.ugr.es

Óscar Cordón
Department of Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
ocordon@decsai.ugr.es

Luis Martínez
Computer Science Department
University of Jaén
Jaén, Spain
martin@ujaen.es

ABSTRACT

The need of support by users for finding out right items in overloaded search spaces is very important in many activities nowadays. One of the activities in which such support is highly demanded is in tourism because tourists visit new scenic places and want to get the best experiences in a limited time. For supporting such needs the use of Recommender Systems have provided good results, but due to the fact that tourism is usually a social activity tourists visit places in groups and demand items and information everywhere and any time. Therefore, the support demanded to Recommender Systems has evolved to Context-Aware and Group Recommendations Systems that is much more challenging. The group recommendations should satisfy all group members, though most proposals do not guarantee that the group recommendation has a high agreement level amongst the group members. Therefore in this contribution is proposed a location-awareness group recommender system that provide recommendations according to the location context of the group and additionally such recommendations are computed to obtain a high agreement among the group members by using a consensus reaching process. The system is implemented by extending a restaurant recommender system REJA (REstaurants of JAén).

Keywords

Group recommender systems, tourism, consensus reaching processes, group decision making, context awareness

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

1. INTRODUCTION

People's interest in spending their spare time at visiting places for leisure has lead to an economy based on tourism in certain countries, such as those with relevant cities, cultures, religions or natural environment. The exploitation of tourist attractions, specially in cities, makes it necessary to help tourists at choosing among many tourism related choices, as it is the case of restaurants. The overloaded choice space and the limited time that tourists can spend to select a choice that meets their preferences leads to a sub-optimal final selection. To overcome this limitation, recommender systems (RS) arose as a successful tool for supporting tourists in their choices with a personalization process by filtering the items according to their interests and needs. Therefore, the RS recommends a reduced set of relevant items to the user.

Classically RS address recommendations about items to individuals. However, there are items such as, restaurants, travels, etc., that have a social component and they are usually enjoyed by groups of people. Group recommender systems (GRS) aims groups of users at finding interesting items among a set of overloaded choices that satisfied the group preferences. There are different approaches to generate the group recommendations [1]. Regardless the technique used, the aim of group recommendations is to satisfy all members and minimise their possible disagreement regarding the recommended products. The basic approaches to produce recommendations without members' disliked items are the *least misery* [2] and *average without misery* [3] methods. Although these methods achieve fairness, they do not guarantee a high level of agreement among the group members over the recommendation. Therefore, our aim in this contribution is to produce group recommendations that not only satisfy members preferences but also have a high degree of agreement.

To increase the agreement of recommendations it is studied the processes of Group decision making (GDM) problems in which agreed solutions are obtained by applying consensus reaching processes (CRP) [4, 5]. A CRP introduces a negotiation process in which the experts modify their initial preferences to bring them closer to the group. Therefore our proposal will apply a consensus-based recommendation approach [6] to achieve a high agreement on the group recommendations.

Additionally to the agreement on recommendations, tourists demand recommendations adapted to their current situation. In these cases, context-aware recommender systems (CARS) [7] are a trend of RS that focuses on delivering recommendations tailored not only to the users' preferences, but also to the circumstances in which the recommendation is requested. Therefore, it is necessary to include their context in order to improve recommendations, in our proposal is used as context the *location* though other context could be included.

Eventually, our proposal of a consensus location-aware recommendation will be integrated in the RS REJA (REstaurants of JAén) [8, 9, 10, 11], a system that recommends restaurants of the province of Jaén, that will combine a consensus driven group recommendation approach [6] with a location-awareness process in order to improve the satisfaction with group recommendations and increase the utility of the recommendations.

The remainder of the paper is structured as follows. First, Section 2 reviews the required background for our proposal. Section 3 describes REJA and extends it to provide context-aware and agreed group recommendations. Finally, Section 4 concludes the contribution.

2. PRELIMINARIES

This section reviews several basic concepts about Recommender Systems, Group Recommender Systems, Context-Aware Recommender Systems, Group Decision Making and Consensus Reaching Processes that are necessary to understand the performance of our proposal.

2.1 Recommender systems and Group recommender systems

A Recommender System (RS) can be described as *"any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options"* [12]. RSs have two main tasks: (i) to gather information about the users, the items and users' needs and interests over the items, and (ii) to recommend products in a personalised way to the users, taking into account the users' preferences.

Formally, the recommendation problem can be defined as finding the most useful item (or set of most useful items) among a large set of choices. To find the best item, a prediction function is approximated by the RS:

$$Recommendation(I,u) = \arg\max_{i_k \in I}[Prediction(i_k,u)] \qquad (1)$$

To obtain the recommendations, the RSs may use information over the users ($U = \{u_1,\ldots,u_m\}$), the items ($I = \{i_1,\ldots,i_n\}$) and the users' ratings over a set of items ($R \subseteq U \times I \rightarrow D$), among other. Depending on how the information is used to recommend, there are different types of RSs:

- Demographic RS [13]. This kind of RS relies on users' demographic attributes, such as the age, gender, or zip code. Most of these systems categorise users regarding their personal information and make recommendations based on the user's class.
- Content-based RS (CBRS) [14]. CBRSs rely on items' information, which can be a textual description or metadata (items' features) [15]. They also need users' feedback over the items and they recommend items that are similar to the ones that the user already experienced and/or liked.
- Knowledge-based RS (KBRS) [16]. In KBRS, the system holds and uses any kind of additional knowledge, such as a user model created from some items that are given as an example of a good item [17], a tweak over the features of a given recommendation (critique-based), or domain specific knowledge that describes items' features and their relations (ontology-based)
- Collaborative filtering RS (CFRS) [18]. Among the different types of RSs, the most successful approach is CFRS, which analyse users' preferences to recommend. This feature makes them able to recommend complex items, because they do not need any item knowledge to produce high quality recommendations.

Due to the fact that our proposal targets recommendations for groups of users and in RS the recommendations are tailored to individuals, it is necessary the use of Group Recommender Systems (GRS) that extends traditional RS to recommend to a target group of users ($G = \{g_1,\ldots,g_r\}$) whose members can have different or even conflicting preferences [19].

In group recommendations, as stated by Jameson in [19], there exist four basic recommending subtasks: (i) acquiring members'

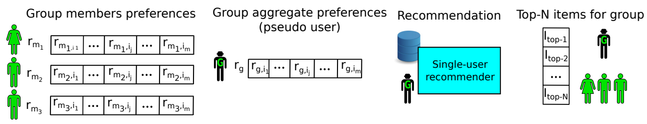


Figure 1: Rating aggregation approach for group recommendation.

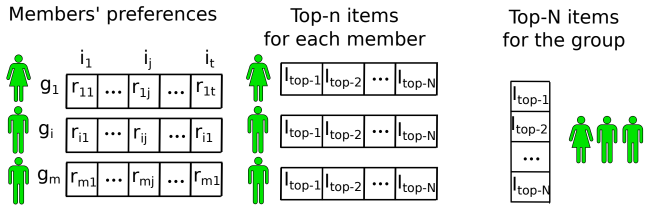


Figure 2: Recommendation aggregation approach for group recommendation.

preferences, (ii) generating the recommendations, (iii) explaining group recommendations, and (iv) aiding to make the final choice. Formally, a GRS tries to find the item (or set of items) that maximises the prediction for a group of users among a set of available items, similarly to individual RS (see Eq. 2).

$$Recommendation(I,G) = \arg\max_{i_k \in I}[Prediction(i_k,G)] \qquad (2)$$

A widespread approach to generate the group recommendations is to apply a single user RS and aggregate the information to produce the group recommendation [1]. Two approaches have been considered in the literature:

- Rating aggregation (see Fig. 1). A group profile is generated from the members' preferences by aggregating them. This pseudo-user profile represents the group preferences and it is used as input of a single user RS to produce the recommendations targeted to the group.
- Recommendation aggregation (see Fig. 2). For each group member it is generated a recommendation. These recommendations are aggregated to produce a single one, which is the recommendation targeted to the group.

Our proposal will use a recommendation aggregation approach that needs to use a single user RS to produce the individual recommendations, in the proposal is used the user-based k-nearest neighbours (UBNN) [20], which recommends items by looking for relations between users' preferences to predict unknown users' ratings (see Fig. 3) according to the following phases: (i) The similarity between the target user and each other system's users is computed, (ii) the most similar users are selected to form the neighbourhood of the target user, (iii) the neighbours' ratings are aggregated to predict the rating for the target user over all unseen or not experienced items, and (iv) the items with the highest prediction are recommended.

2.2 Context-aware recommender systems

Previous RSs assume that the user's satisfaction towards the recommendations is only dependent of his/her preferences, thus finding the best item or set of items can be done by analysing the ratings solely. But in some scenarios, the user's satisfaction with a given recommendation can depend on the items recommended and also other factors, such as the time when the recommendation is requested, the item's location, or the user's circumstances.

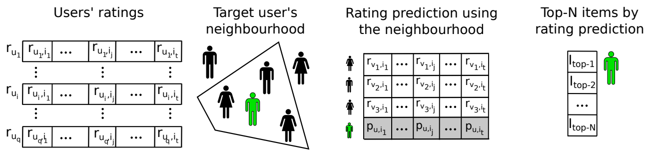


Figure 3: Single user recommendation through user based collaborative filtering.

Context-aware recommender systems (CARS) [7] are an extension of traditional RS that include contextual information in the recommendation calculus. The contextual information describes the context in which the recommendation is requested or presented to the user.

Several alternatives to include context-awareness in a RS have been proposed. These proposals can be classified into three approaches [7]:

- Contextual pre-filtering. It selects only the users' ratings that were generated in the target user's context.
- Contextual post-filtering. It modifies the ratings prediction regarding their suitability on the target user's context.
- Contextual modelling. The context is used in the prediction function as an input, added to the target user and item.

Researchers have found that none of these approaches completely dominates the other ones [7], therefore a study on the concrete system must be done to determine the best approach on the target recommendation scenario.

In tourism RS, different authors have pointed out the relevance of different contextual dimensions [21]. An important contextual information is the users' and items' location, which can influence the prediction or filter out items that are too far to reach from the user's location. Other important contextual dimension is the time, given that the recommendation of a set of tourist activities should be different if it is for summer or for winter. Other relevant contextual information are the weather, local time, user's mood, or companion, among others. In our proposal the context will be defined by the location of users and items.

2.3 Consensus Reaching Processes in Group Decision Making

In Group Decision Making (GDM) [22] problems, a set of experts ($E = \{e_1, \ldots, e_p\}$) tries to find the best solution among a set of alternatives ($A = \{a_1, \ldots, a_q\}$). There are different contexts in which the decision can occur, such as certainty, risk, and uncertainty. Most of the decisions in the real world occur in uncertainty context. To manage the uncertainty, the most used structure is a fuzzy preference relation.

A fuzzy preference relation [23] P_i given by an expert e_i is defined by a membership function $\mu_{P_i} : A \times A \to [0,1]$. This function is represented by a matrix of size $q \times q$, and each μ_i^{kl} denotes $\mu_{P_i}(a_k, a_l)$, this is, the preference degree of the alternative a_k over a_l, regarding the expert e_i. This preference degree can be less, equal, or greater than 0.5, indicating the degree to which a_k is preferred, are indifferent, or the degree to which a_l is preferred, respectively.

Once the experts have expressed their individual preferences, a selection process is performed to obtain a solution set of alternatives. However, this process does not guarantee an agreement on the solution and then experts might feel that their opinion has been overlooked or even that they reject the selected solution. To avoid

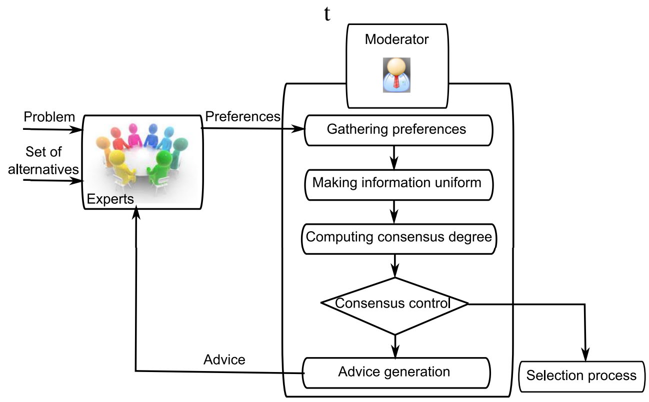


Figure 4: Scheme of resolution of a group decision making problem with a consensus reaching process.

the previous problem, Consensus Reaching Processes (CRP) [24] were introduced in GDM to achieve agreed solutions. It means that there exists a mutual agreement between the group member and each individual opinion has been taken into account to maximise the group satisfaction [25]. A CRP aims to reach a given agreement level before making the final selection of the alternative by means of an iterative discussion process among experts until they meet the consensus condition (see Fig. 4) [26]:

- Consensus measure: Using the preferences of each expert, the consensus degree of the group $cr \in [0,1]$ is calculated.
- Consensus control: Being μ the consensus degree required, it is checked if $cr > \mu \in [0,1]$. If it does, the consensus degree meets the requirement and the process ends. To avoid that the CRP takes too many rounds, a maximum number of rounds can be established. This finalises the process although the consensus degree required has not been reached.
- Consensus progress: If the consensus degree required has not been reached, the moderator communicates to each expert the preference modification that they should consider to reach the consensus degree.

A CRP is usually supervised by a moderator through the following functions, which in some cases can be automated [27]:

- Assess the agreement level of the experts.
- Find alternatives far from consensus.
- Advice the experts the preference changes that they should consider to increment the consensus.

2.4 Consensus driven group recommendation

Due to the fact, that this contribution aims at obtaining agreed recommendations for groups, it will implement the consensus-driven GRS approach [15] that applies an automatic CRP [28] in the recommendation aggregation process to improve the satisfaction of the members towards the group recommendation. The general scheme of the consensus-driven GRS follows these phases:

- Individual recommendation phase: First, the system uses the individual ratings over the restaurants to produce a recommendation tailored to each member.
- Consensus phase: An automatic consensus reaching process is applied to the individual recommendations. This process updates the individual recommendations in several iterations

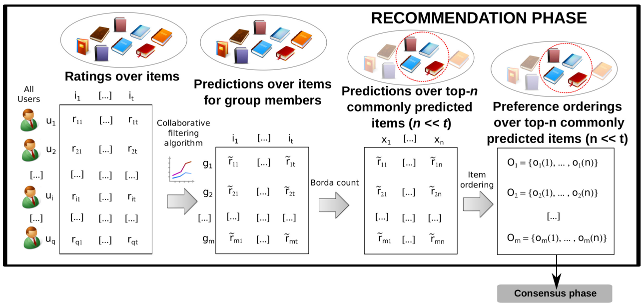


Figure 5: Recommendation phase scheme.

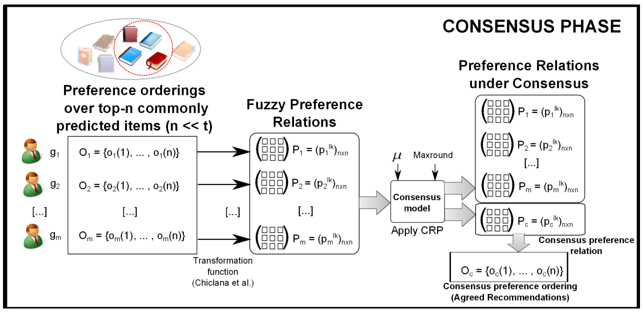


Figure 6: Consensus phase scheme.

until the consensus degree reaches an acceptable level and generates the collective recommendation.

These phases are described in further detail in the remaining of this section.

2.4.1 Individual Recommendation phase

In the individual recommendation phase (see Fig. 5), members' recommendations are computed using a single user RS, which produces an ordered list of items for each member. A subset of items is selected from the set of items recommended to all members. The orderings that the subsequent CRP uses are given on the subset selected.

Specifically, the recommendation phase is composed of the following steps:

1. The individual recommendations for each member are generated in the individual recommendation phase. To do so, a single user RS predicts the rating of unseen items for each member:
$$\tilde{r}_{g_i i_k} = Prediction(g_i, i_k) \qquad i_k \in \{i_l \in I \mid \forall g_i \nexists r_{g_i i_k}, g_i \in G\} \tag{3}$$
2. Once the predictions are generated, it is needed to take into account that, for some items, it might not be possible to predict a rating. For this reason, we consider only the items for which the system is able to produce a prediction for all the group members. Therefore, a subset of items is built and only the items in I^G set are taken into account in the next phase:
$$I^G = \{i_k \in I \mid \forall g_i \exists \tilde{r}_{g_i i_k}, g_i \in G\} \tag{4}$$
3. A total order of the items in I^G set is obtained for each group member regarding the prediction value:
$$O_{g_i} = \{o_{g_i}(i_1), \ldots, o_{g_i}(i_k), \ldots, o_{g_i}(i_s)\}, \quad i_k \in I^G \tag{5}$$
4. A reduced subset of items $I_t^G \subseteq I^G$ is built, composed of the t best products for the group using the Borda count over all members O_{g_i}. A total order $\tilde{O}_{g_i}$ over I_t^G set is built for each member, keeping the same order that the items had in O_{g_i}:
$$\tilde{O}_{g_i} = \{\tilde{o}_{g_i}(i_1), \ldots, \tilde{o}_{g_i}(i_k), \ldots, \tilde{o}_{g_i}(i_t)\}, \quad i_k \in I_t^G \tag{6}$$

2.4.2 Consensus phase

In the consensus phase (see Fig. 6), the individual recommendations of the members are combined to produce the group recommendations. A CRP then tries to obtain an agreed recommendation list for the group. This is done by applying an automatic CRP, which generates a recommendation list with a high consensus level among the members.

Specifically, the consensus phase is composed of the following steps:

1. Each total ordering $\tilde{O}_{g_i}$ is transformed into a fuzzy preference relation by using the following equation [29]:
$$p_{g_i}^{lk} = \frac{1}{2}\left(1 + \frac{\tilde{o}_{g_i}(i_k) - \tilde{o}_{g_i}(i_l)}{t-1}\right), \quad i_k, i_l \in I_t^G \tag{7}$$
where $\tilde{o}_{g_i}(i_k)$ and $\tilde{o}_{g_i}(i_l)$ are the position of items i_k and i_l for user g_i, respectively. An example is provided in order to clarify the behaviour of eq. (7). Let $I_t^G = \{i_1, i_2, i_3, i_4, i_5\}$ and $\tilde{O}_{g_1} = \{1, 5, 4, 3, 2\}$. The fuzzy preference relation for member g_1 is:
$$P_{g_1} = \begin{pmatrix} 0.5 & 1 & 0.88 & 0.75 & 0.63 \\ 0 & 0.5 & 0.38 & 0.25 & 0.13 \\ 0.13 & 0.63 & 0.5 & 0.38 & 0.25 \\ 0.25 & 0.75 & 0.63 & 0.5 & 0.38 \\ 0.38 & 0.88 & 0.75 & 0.63 & 0.5 \end{pmatrix}$$
where $p_{g_i}^{12} = 1$ indicates that item i_1 is totally preferred to i_2. The fuzzy preference relation is symmetric, hence $p_{g_i}^{21} = 0$ and indicates the same. When $p_{g_i}^{kl} = 0.5$, both items are equally preferred. When $i_k = i_l$ the preference is also 0.5, which is shown in the main diagonal of P_{g_1}.
2. Once the fuzzy preference relations are generated, an automatic CRP is applied over them. The CRP is composed of the following phases [26]:
   - Consensus measure: The similarity matrix of each pair of members is obtained from the similarity between their fuzzy preference relationships.
$$SM_{g_i g_j} = (sm_{g_i g_j}^{i_k i_l})_{t \times t} \tag{8}$$
$$sm_{g_i g_j}^{i_k i_l} = 1 - |(p_{g_i}^{i_k i_l} - p_{g_j}^{i_k i_l})| \tag{9}$$
After this, the group's consensus matrix is generated from all the similarity matrices of all members:
$$CM = (cm^{i_k i_l})_{t \times t} \tag{10}$$
$$cm^{i_k i_l} = OWA_W(\cup_{g_i g_j} sim_{g_i g_j}^{i_k i_l}) \tag{11}$$
where OWA_W is an Ordered Weighted Average operator [30] whose behaviour is determined by W.
With CM matrix, $cr \in [0, 1]$, is computed, which is the consensus level of the group:
$$cr = \sum_{i_k \in I_t^G} \frac{ca^{i_k}}{t} \tag{12}$$

$$ca^{i_k} = \sum_{i_l \in I_t^G - \{i_k\}} cm^{i_k, i_k} \qquad (13)$$

- Consensus control: In this step it is checked if $cr \geq \mu \in [0,1]$, being μ the required consensus degree. If $cr \geq \mu$, the consensus is reached and the collective preference is generated.
- Advice generation: If the consensus level has not reached the required consensus degree, the individual preferences are updated in a way that the further preferences of the group are modified automatically to bring them closer to the group preference. Specifically, the advice generation is done in the following way:
  - The collective preference P_G is computed.
  - The proximity matrix PP_{g_i} between each member g_i and P_G is computed:

$$pp_{g_i}^{i_k i_l} = 1 - |(p_{g_i}^{i_k i_l} - p_G^{i_k i_l})| \qquad (14)$$

  - The pairs of items whose consensus degrees ca^{i_k} and $cr^{i_k i_l}$ are not enough are identified:

$$CC = \{(i_k, i_l) | ca^{i_k} < \mu \wedge cr^{i_k i_l} < \mu\} \qquad (15)$$

  - The experts whose preferences over the pairs in CC should change are identified by checking if the proximity of the expert is lower than the average proximity.
  - The modified preferences are computed using the following equation:

$$\tilde{p}_{g_i}^{i_k i_l} = \begin{cases} \max(p_{g_i}^{i_k i_l} + 0.1, 1) & if \quad p_{g_i}^{i_k i_l} < p_G^{i_k i_l} \\ p_{g_i}^{i_k i_l} & if \quad p_{g_i}^{i_k i_l} = p_G^{i_k i_l} \\ \min(p_{g_i}^{i_k i_l} - 0.1, 0) & if \quad p_{g_i}^{i_k i_l} > p_G^{i_k i_l} \end{cases} \qquad (16)$$

3. When the CRP ends, a fuzzy preference relationship with a high consensus is obtained. Then, the group recommendation is computed using the non-dominance degree of each alternative that expresses to what extent an alternative is non-dominated by the rest [31]:

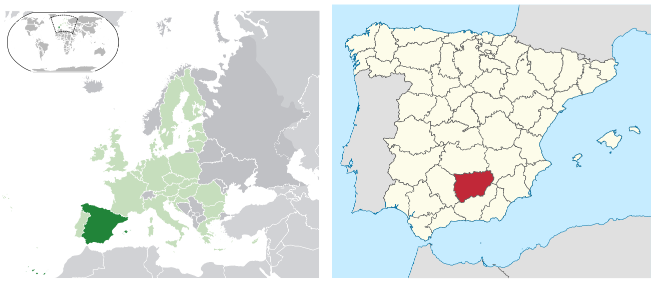
$$p^{ND}(i_k) = 1 - \sup_{i_l \in I_t^G} p^s(i_l, i_k) \qquad (17)$$

where $p^{ND}(i_k)$ is the non-dominance degree of item i_k and $p^s(i_l, i_k)$ is:

$$p^s(i_k, i_l) = \begin{cases} p(i_k, i_l) - p(i_l, i_k) & if \quad p(i_k, i_l) \leqq p(i_l, i_k) \\ 0 & otherwise \end{cases} \qquad (18)$$

3. CLG-REJA: A CONSENSUS LOCATION-AWARENESS GROUP RECOMMENDER FOR RESTAURANTS

This section introduces the consensus location-awareness group recommender scheme that is implemented on an app restaurant recommender system REJA. Therefore, first it is described the basic data and performance of REJA and later on it is described the performance of the location awareness and consensus recommendations to conclude with the interface of the app that can be used to obtain such a type of recommendations.



(a) Spain (b) Province of Jaén

Figure 7: Province of Jaén, area of interest of REJA

3.1 Restaurants of Jaén Recommender System: REJA

Even though there are different alternatives to search and check restaurants using widespread applications such as Yelp or TripAdvisor. However, location specific applications provide an added value over general ones. REJA[1] (REstaurants of JAén) [8, 9, 10, 11] is a system developed by Sinbad[2] Research Group at the University of Jaén (Spain) and it is focused on the recommendation of restaurants located in the province of Jaén.

Before describing REJA, it is interesting to provide some data about the environment of this system. The province of Jaén population is 664,916, distributed in 13,496km^2. The most important economic activity is olive oil production, which occupies around 80% of the cultivable land. Other important features related to tourism are that it has 4 nature parks, the preservation of a number of castles in different towns, and the preservation of a number of renaissance monuments, such as churches and palaces. This makes that, added to other tourism facilities, there are a number of restaurants distributed along the province.

REJA is a system that supports users at finding restaurants in the province of Jaén. It relies on explicit ratings over the restaurants. The restaurant database has 516 restaurants and holds additional information over them such as location, phone number, type of cuisine, and other relevant information over the restaurant facilities.

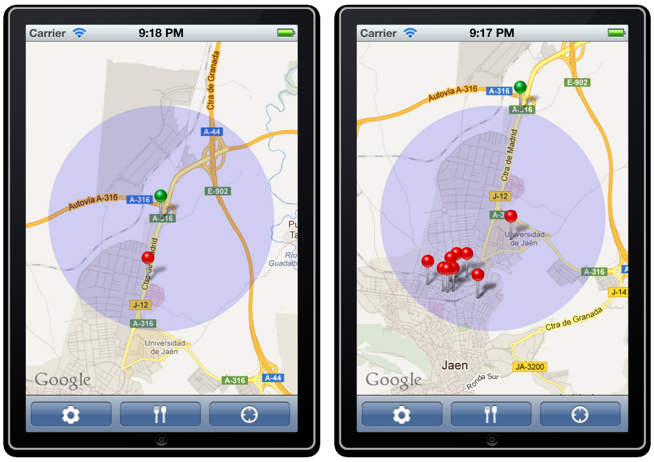
It may provide recommendations for anonymous users, REJA produces non-personalised recommendations such as most-liked and most popular restaurants and also enables the search of similar restaurants to a given one. However for registered users, REJA provides ***collaborative recommendations*** (CFRS). To obtain recommendations, a registered user must provide enough ratings about the restaurants known by her (at least 20 ratings). This information is used to build and modify the user's profile and to compute suitable recommendations for her.

When REJA is used by a user with a small amount of information, such as a novel user, CFRS face the problem of cold-start, which makes that the system cannot generate the recommendations or it produces low quality ones. To overcome this limitation and produce recommendations for such users, it implements a commuted hybrid recommender system [9] that hybridize the former CFRS and a knowledge-based system.

3.2 Including context awareness for recommendations on the move.

The previous functionalities of REJA [9] are targeted to users that interact with the system through a web interface at home. How-

[1] `http://sinbad2.ujaen.es/reja`



(a) On site request (b) On-the-move request

Figure 8: Area of interest for different user's contexts.

ever, users' interaction with the systems is done mostly through mobile devices in spite of their limitations such as screen size and battery duration. However, most of them have built-in sensors, such as barometer, accelerometers, wireless communication interfaces, compass and Global Positioning System (GPS), that can automatically gather information, which simplifies user's interaction. For this reason, users' interaction through mobile devices provide valuable information, which can be used to produce recommendations tailored to the specific user's context. Thus, REJA was extended to allow users access through mobile devices and enabled the possibility of users requesting restaurant recommendations to REJA, with the *location-awareness* requirements. So REJA integrates a CARS that takes into account the user's location and speed [10] that are used in a fuzzy system to adjust the parameters of an area of interest.

In the example depicted in Fig. 8 the user's location is represented with the green pin. As she is travelling to the city, the restaurants that he already left behind her are no longer interesting, and makes all the restaurant ahead a better option than the restaurants that the user has already left behind.

3.3 Location-awareness and consensus driven group recommendation

So far, REJA recommends restaurants for individual users using different approaches, such as hybridised or context-aware recommendation, among others. However, as it has been pointed out restaurant are social items enjoyed generally by groups. Therefore the restaurant recommender systems are used by groups. For this reason, this contribution adds to the location-aware REJA system a group recommendation approach based on consensus, not only to cope with the social requirement but also to provide highly agreed recommendations that provide satisfaction to the whole group.

Therefore, in the process of generating the group recommendations it is necessary to adjust them to the specific group's context. In the case of REJA, the context considered is the location of the different items and the position of the group members. From the three approaches to integrate context-awareness into a recommender system (see section 2.2). The approach used in this proposal is *contextual post-filtering*, which allows to filter and re-rank the items after they are recommended according to the items' and the user's context. Given that the context considered in REJA is the location of the different items and the position of the group members. Therefore, the items far from the users are penalised.

To integrate location-awareness in the consensus-driven GRS [6], the system has been modified to include the group's context. For this reason, in the integration of the model in CLG-REJA it is necessary to include a contextualisation phase. Therefore, the scheme for the consensus location-awareness group recommender system is composed of the three following phases (see Fig. 9):

1. Individual recommendation phase: The system generates the members' individual recommendations using a single user RS.
2. Recommendation contextualisation phase: The recommendations are post-filtered to incorporate the location information and produce localised individual recommendations.
3. Consensus phase: The contextualised individual recommendations are fed to the automatic consensus module that produces the group recommendations.

The phases of the system are described in further detail in the remaining of this section.

3.3.1 *Individual recommendation phase*

The individual recommendations for each member are generated in the individual recommendation phase. To do so, a single user RS produces a list of items, which are new for all members, sorted by their rating prediction (see Eq. 3).

Once the predictions are generated, it is needed to take into account that, for some items, it might not be possible to predict a rating for all members. These items are excluded from the recommendation (see Eq. 4).

3.3.2 *Recommendation contextualisation phase*

The individual recommendation phase output is the predictions for all the items with a prediction for all group's members. In this phase, these predictions are modified to exclude elements that are far from the group's location, therefore the items are re-ranked regarding their distance.

The items' re-ranking is performed by using a fuzzy method to allow certain flexibility. Thus, the group manager needs to establish a parameter δ, which is the distance that the group is willing to move to reach the item recommended. With this information, the system modifies the predictions of the items to discard those that are too far to reach, maintains the predictions of the items that lie within δ and modifies in a soft way the items that lie outside but near of δ:

Therefore, a modification is applied to the prediction of each item regarding their respective distance to the group:

$$\tilde{r}'_{g_i,i_k} = r_{g_i,i_k} * w_{G,i_k} \qquad , w_{G,i_k} \in [0,1] \tag{19}$$

$$w_{G,i_k} = \begin{cases} 1 & if \quad d(G,i_k) \leq \delta \\ 1 - \frac{d(G,i_k)-\delta}{\delta'-\delta} & if \quad \delta \leq d(G,i_k) \leq \delta' \\ 0 & if \quad d(G,i_k) \geq \delta' \end{cases} \tag{20}$$

where $d(G,i_k)$ is the distance between the group and the item, δ is defined by the group manager, and δ' value is defined from δ:

$$\delta' = \delta * (1+\alpha), \qquad \alpha \in [0,1] \tag{21}$$

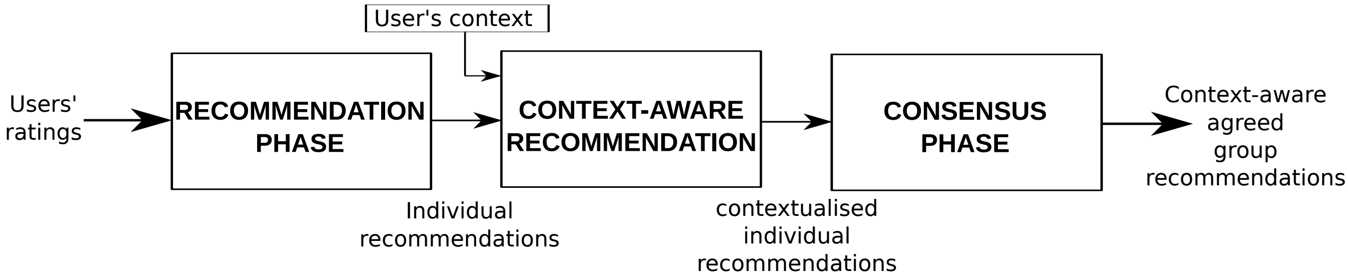


Figure 9: General scheme of the context-aware group recommender of REJA

where α is a parameter that defines how flexible is the contextual filtering. In REJA, α is set to 0.2, but it might be different in other recommendation domains.

After the contextualisation is done, it is needed to transform the contextualised predictions to a preference relation, in order to be used in the consensus phase. Similarly to the consensus-driven GRS, the total order for each member is obtained regarding the contextualised prediction value (r'_{g_i,i_k}):

$$O'_{g_i} = \{o'_{g_i}(i_1), \ldots, o'_{g_i}(i_k), \ldots, o'_{g_i}(i_s)\}, \quad i_k \in I^G \tag{22}$$

3.3.3 *Consensus phase*

The preference relation obtained in the previous phase describes the members' initial preferences over the items. However, to use them in a CRP, the number of items must be reduced. This reduction is done using the Borda count to select t items and compose I_t^G. After that, $\tilde{O}'_{g_i}$ are built maintaining the order in $\tilde{O}_{g_i}$:

$$\tilde{O}'_{g_i} = \{\tilde{o}'_{g_i}(i_1), \ldots, \tilde{o}'_{g_i}(i_k), \ldots, \tilde{o}'_{g_i}(i_t)\}, \quad i_k \in I_t^G \tag{23}$$

The total orderings $\tilde{O}'_{g_i}$ are transformed into fuzzy preference relationships [29] using the following equation:

$$p_{g_i}^{lk} = \frac{1}{2}\left(1 + \frac{\tilde{o}'_{g_i}(i_k) - \tilde{o}'_{g_i}(i_l)}{t-1}\right), \quad i_k, i_l \in I_t^G \tag{24}$$

Once the fuzzy preference relations are generated, an automatic CRP [24] is applied over them, as explained in steps 2 and 3 of the Consensus phase described in section 2.4.2

3.4 A Consensus Location-awareness group recommendation app for REJA

An operational prototype that implements the system described in section 3.3 has been developed with the aim at studying the performance of our proposal under real world contextual conditions. Our prototype aims at providing group restaurant recommendations in the province of Jaén.

The architecture of the prototype (see Fig. 10) follows the client-server paradigm that comprises two elements: the mobile clients and the remote server. On one hand, the mobile clients consists of a mobile application that is installed on the mobile devices. The application is in charge of creating the group, gathering the contextual knowledge, provide the server the group's information, and display the group recommendations. On the other hand, the remote server provides a web service that allows to the group creator to send the group's information to the server and request group recommendations generated with the system described in section 3.3.

Therefore, the users of the system are required to install a mobile application on their devices. Once the application is launched,

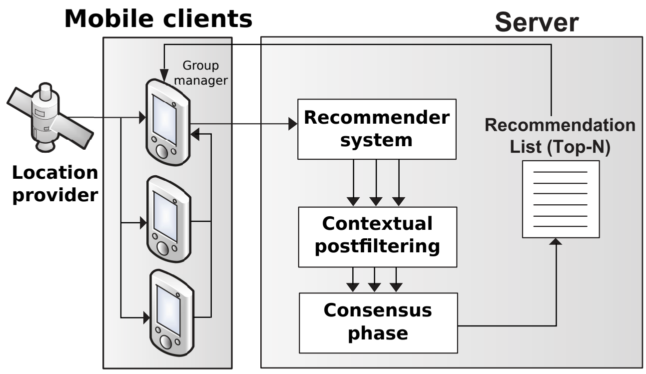


Figure 10: Architecture of the prototype for location-aware consensus-driven group recommendations.

the users are requested to provide their log-in data. Figure 11 depicts the log-in interface of the prototype and the initial screen for a logged user.

After this task is completed by all the group members the group that later on wants to request the recommendation is created. The aim of this task is to specify the users that belong to the group. Figure 12 illustrates how the prototype allows group creation. To perform this task minimising the group's members interaction with their mobile devices, the specification of the group members is done by the user with a special user role, the *group creator*. therefore, the group creator, added to the creation of the group, has the task of adding members to the group.

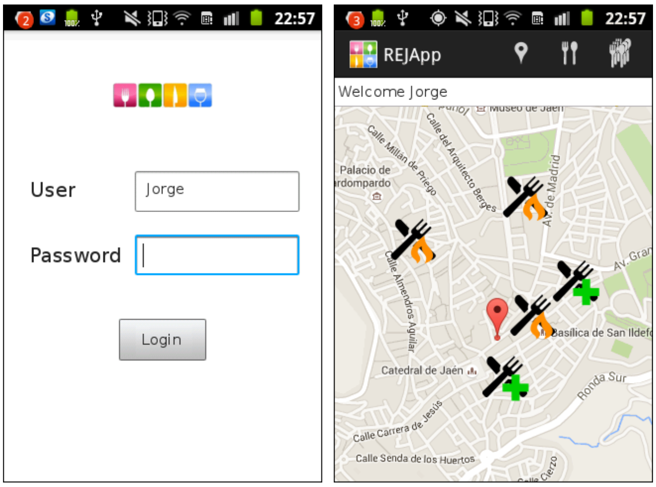
Once the group creation is done, the system allows the group creator to request the recommendations (see Fig. 13a). Before the actual group recommendation request, the group creator needs to express how far the group is willing to move to reach a good restaurant. For this, the interface provides a slider in which the group creator picks the desired value for δ.

When the system generates the recommendations, the mobile device presents the recommended items in the map, together with the group location (see Fig 13b). The map visualisation allows them to make the final decision taking into account the closeness of the restaurants recommended.

4. CONCLUDING REMARKS

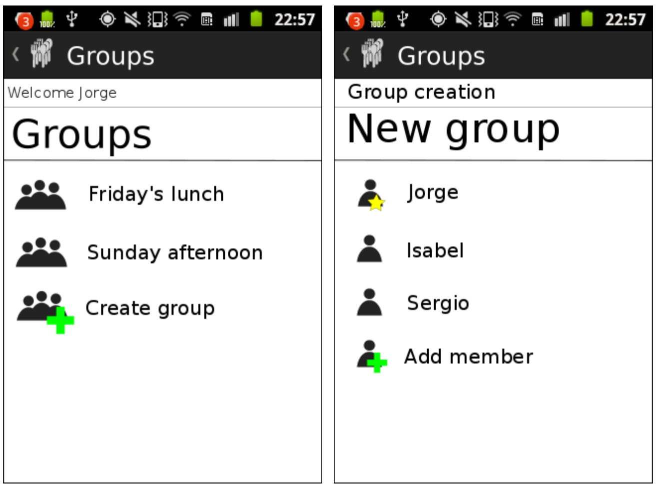
In this contribution, the improvement of classical recommender system for tourist purposes has been considered, taking into account two important issues within tourism namely, the ubiquity and social feature that involves tourism activities.

Therefore, a general recommendation scheme has been introduced, which is able to deal with context awareness and agreed group decisions. It has been implemented in a restaurant RS so-



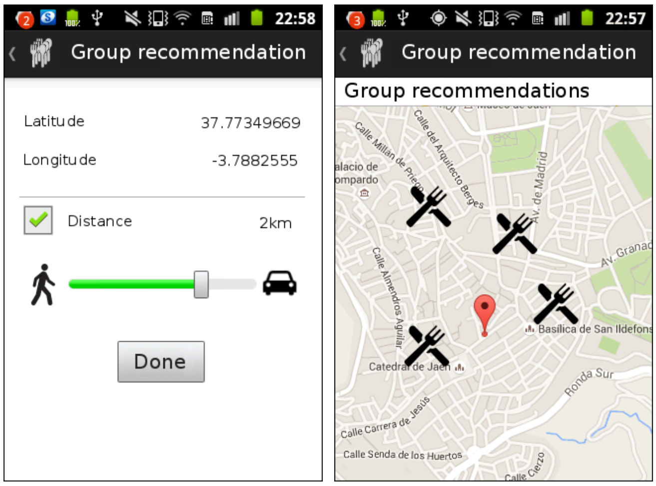
(a) Login page. (b) Logged user screen.

Figure 11: Screens of the prototype for the login task.



(a) Group creation. (b) Members specification.

Figure 12: Screens of the prototype for the group creation and members specification tasks.



(a) Recommendation request. (b) Map visualisation.

Figure 13: Screenshots of the recommendation request and visualisation.

called REJA by means of a mobile app.

As future work, we plan to develop a study of how the users perceive the utility of this kind of recommendation compared to others. Also we plan to develop an user study to evaluate the interaction of the users with the system and the satisfaction with the recommendations.

Other interesting future work is to integrate additional contextual dimensions additionally to the location, such as the climate or the week-day. These contexts are particularly interesting given that certain restaurant's facilities could change their influence on users' satisfaction in certain contexts. An example of this situation might be the availability of a terrace in a rainy day on winter.

Acknowledgements:

This research work was partially supported by the Research Project TIN-2012-31263 and the Spanish Ministry of Education, Culture and Sport FPU fellowship (FPU13/01151).

5. REFERENCES

[1] J. Masthoff, "Group recommender systems: Combining individual models," in *Recommender Systems Handbook*. Springer US, 2011, ch. 21, pp. 677–702.

[2] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl, "Polylens: a recommender system for groups of users," in *Proceedings ECSCW'01*, 2001, pp. 199–218.

[3] J. Masthoff and A. Gatt, "In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems," *User Modelling and User-Adapted Interaction*, vol. 16, no. 3-4, pp. 281–319, SEP 2006.

[4] I. Palomares, L. Martínez, and F. Herrera, "A consensus model to detect and manage noncooperative behaviors in large-scale group decision making," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 516–530, 2014.

[5] I. Palomares and L. Martínez, "A semi-supervised multi-agent system model to support consensus reaching processes," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 762–777, 2014.

[6] J. Castro, F. J. Quesada, I. Palomares, and L. Martínez, “A consensus-driven group recommender system,” *International Journal of Intelligent Systems*, vol. 30, no. 8, pp. 887–906, 2015.

[7] G. Adomavicius and A. Tuzhilin, “Context-aware recommender systems,” in *Recommender Systems Handbook*. Springer US, 2011, ch. 3, pp. 217–253.

[8] L. Martínez, R. M. Rodríguez, and M. Espinilla, “Reja: A georeferenced hybrid recommender system for restaurants,” in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, ser. WI-IAT ’09. IEEE Computer Society, 2009, pp. 187–190.

[9] R. M. Rodríguez, M. Espinilla, P. J. Sánchez, and L. Martínez-Lopez, “Using linguistic incomplete preference relations to cold start recommendations,” *Internet Research*, vol. 20, no. 3, pp. 296–315, 2010.

[10] J. Noguera, M. Barranco, R. Segura, and L. Martínez, “A mobile 3d-gis hybrid recommender system for tourism,” *Information Sciences*, vol. 215, pp. 37–52, 2012.

[11] M. Barranco, J. Noguera, J. Castro, and L. Martínez, “A context-aware mobile recommender system based on location and trajectory,” in *Management Intelligent Systems*, ser. Advances in Intelligent Systems and Computing, J. Casillas, F. J. Martínez-López, and J. M. Corchado Rodríguez, Eds. Springer Berlin Heidelberg, 2012, vol. 171, pp. 153–162.

[12] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modelling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[13] M. J. Pazzani, “A framework for collaborative, content-based and demographic filtering,” *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[14] P. Lops, M. Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in *Recommender Systems Handbook*. Springer US, 2011, ch. 3, pp. 73–105.

[15] J. Castro, R. M. Rodríguez, and M. J. Barranco, “Weighting of features in content-based filtering with entropy and dependence measures,” *International Journal of Computational Intelligence Systems*, vol. 7, no. 1, pp. 80–89, 2014.

[16] S. Trewin, “Knowledge-based recommender systems,” *Encyclopedia of Library and Information Science: Volume 69-Supplement 32*, 2000.

[17] L. Martínez, M. J. Barranco, L. G. Pérez, and M. Espinilla, “A knowledge based recommender system with multigranular linguistic information,” *International Journal of Computational Inteliggence Systems*, vol. 1, no. 3, pp. 225–236, 2008.

[18] Y. Koren and R. Bell, “Advances in collaborative filtering,” in *Recommender Systems Handbook*. Springer US, 2011, ch. 5, pp. 145–186.

[19] A. Jameson and B. Smyth, “Recommendation to groups,” in *The Adaptive Web*, ser. Lecture Notes in Computer Science, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Springer Berlin Heidelberg, 2007, vol. 4321, pp. 596–627.

[20] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens: an open architecture for collaborative filtering of netnews,” in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. New York, USA: ACM, 1994, pp. 175–186.

[21] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, “Mobile recommender systems in tourism,” *Journal of Network and Computer Applications*, vol. 39, no. 0, pp. 319 – 333, 2014.

[22] J. Lu, G. Zhang, D. Ruan, and F. Wu, *Multi-Objective Group Decision Making*. Imperial College Press, 2006.

[23] R. Parreiras, P. Ekel, and F. B. Jr., “A dynamic consensus scheme based on a nonreciprocal fuzzy preference relation modeling,” *Information Sciences*, vol. 211, no. 1, pp. 1–17, 2012.

[24] Z. Xu, “An automatic approach to reaching consensus in multiple attribute group decision making,” *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1369–1374, 2009.

[25] S. Saint and J. R. Lawson, *Rules for Reaching Consensus. A Modern Approach to Decision Making*. Jossey-Bass, 1994.

[26] E. Herrera-Viedma, F. J. Cabrerizo, J. Kacprzyk, and W. Pedrycz, “A review of soft consensus models in a fuzzy environment,” *Information Fusion*, vol. 17, pp. 4–13, 2014.

[27] I. Palomares, F. Estrella, L. Martínez, and F. Herrera, “Consensus under a fuzzy context: Taxonomy, analysis framework AFRYCA and experimental case of study,” *Information Fusion*, vol. 20, no. November 2014, pp. 252–271, 2014.

[28] Z. Wu and J. Xu, “A consistency and consensus based decision support model for group decision making with multiplicative preference relations,” *Decision Support Systems*, vol. 52, no. 3, pp. 757–767, 2012.

[29] F. Chiclana, E. Herrera-Viedma, S. Alonso, and R. Marqués-Pereira, *H. Bustince et al. (eds.), Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*. Springer, 2008, ch. Preferences and Consistency Issues in Group Decision Making, pp. 219–237.

[30] R. Yager, “On ordered weighted averaging aggregation operators in multi-criteria decision making,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[31] S. Orlovsky, “Decision-making with a fuzzy preference relation,” *Fuzzy sets and systems*, vol. 1, no. 3, pp. 155–167, 1978.