

Hybrid Group Recommendations for a Travel Service

Toon De Pessemier · Jeroen Dhondt ·
Luc Martens

Received: date / Accepted: date

Abstract Recommendation techniques have proven their usefulness as a tool to cope with the information overload problem in many classical domains such as movies, books, and music. Additional challenges for recommender systems emerge in the domain of tourism such as acquiring metadata and feedback, the sparsity of the rating matrix, user constraints, and the fact that traveling is often a group activity. This paper proposes a recommender system that offers personalized recommendations for travel destinations to individuals and groups. These recommendations are based on the users' rating profile, personal interests, and specific demands for their next destination. The recommendation algorithm is a hybrid approach combining a content-based, collaborative filtering, and knowledge-based solution. For groups of users, such as families or friends, individual recommendations are aggregated into group recommendations, with an additional opportunity for users to give feedback on these group recommendations. A group of test users evaluated the recommender system using a prototype web application. The results prove the usefulness of individual and group recommendations and show that users prefer the hybrid algorithm over each individual technique. This paper demonstrates the added value of various recommendation algorithms in terms of different quality aspects, compared to an unpersonalized list of the most-popular destinations.

Keywords Recommender system · Group recommendations · Travel · Tourism · Hybrid · Collaborative filtering · Content-based recommender

T. De Pessemier - J. Dhondt - L. Martens
iMinds - Ghent University
G. Crommenlaan 8 / 201, 9050 Ghent, Belgium
Tel.: +32-09-33-14908
Fax: +32-09-33-14899
E-mail: toon.depessemier@ugent.be
E-mail: jeroen.dhondt@ugent.be
E-mail: luc1.martens@ugent.be

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 can help to overcome the problem of information overload. Recommender systems are software tools and techniques providing suggestions for items to be of interest to a user [30]. Over the years, several different approaches for generating recommendations have been proposed [5]. *Knowledge-based* systems generate recommendations using specific domain knowledge about how certain item features meet user preferences, and ultimately, how the item is of interest to the user [11]. A specific type of knowledge-based recommenders are constraint-based recommenders. Constraint-based recommenders exploit predefined knowledge bases that contain explicit rules about how to relate customer requirements with item features [32].

The concept of a *content-based* recommender system is to suggest the items that are similar to the items that the user liked in the past. The recommender system learns which (type of) items the user likes based on the user's consumption behavior and the attributes that describe an item. These attributes provide useful information about the item, such as the title, a description, keywords, categories, etc., and can be denoted as metadata. The items' attributes are used to construct a user profile, in which the personal preferences and interests of the user are stored. Then, the main operation performed by a content-based recommender consists in matching the attributes of the user's profile with the attributes of (unexplored) content items, with the aim of finding interesting recommendations that match the user's preferences [32].

The hypothesis of systems based on *collaborative filtering* is that if users shared the same or similar preferences in the past, they will also have similar interests in the future. The similarity in preferences of two users is calculated based on the similarity in the consumption history of the users. So, if for example, user A and B have expressed similar preferences to the same items in the past, and user A had recently consumed and liked a new item that B has not yet explored, the basic rational is to suggest this new item also to B. Nearest neighbor methods are very popular collaborative filtering solutions because of their simplicity, efficiency, and ability to produce accurate recommendations. In nearest neighbor approaches, the user-item consumptions stored in the system are directly used to predict the user's preferences for new items. This can be done in two ways known as user-based or item-based recommendation [32].

In the context of traveling, recommender systems can provide users valuable recommendations for destinations tailored to their personal preferences, requirements, and constraints. However, most research activities on recommender systems focus on domains such as movies, songs, or items of an online shop. Specific characteristics of the domain make recommendations for travel destinations a lot harder. Firstly, data regarding travel destinations (meta-data) are usually not conveniently available and the user ratings are harder to acquire than the freely available datasets for movies such as MovieLens [20]. 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 [8]. 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.

2 Related Work

The domain of tourism and traveling is very appealing for recommender systems research [3]. The commercial value of tourism is huge and eTourism is becoming increasingly popular. Moreover, people usually spend a considerable amount of time at planning their travel, considering different options, and asking others for advice before making a decision. However, the particular characteristics of this domain provoke the appearance of novel problems and the need of developing new techniques to solve them [27].

The importance of recommender systems in the domain of tourism was already stated by Ricci back in 2002 [31]. He explained that content-based and knowledge-based systems are very suitable for recommending travel destinations. Travel destinations can be portrayed by rather stable concepts, and as such a good knowledge base can be reused by different engines. Case-based reasoning, for instance, has been used to offer users recommendations based on the knowledge of previously made travel plans that are stored in the system [33]. This approach allowed users to alter and tweak these existing travel plans and save them in the system.

Activities such traveling, dining, or enjoying the nightlife are usually social activities, which are carried out in groups of people (couples, families, colleagues, friends); thus, it is necessary to take into account the preferences of all the participants when providing recommendations [27]. This introduced the need for intelligent systems generating group recommendations covering the preferences of all group members.

Various (group) recommender systems for points-of-interest (POIs), such as tourist attractions, restaurants, and hotels, have been proposed in literature. CLG-REJA is an extension of the REJA restaurant recommender [23], which consists of two phases, just like our approach. In the first phase, a list of recommendations for each group member is constructed, taking into account personal ratings. In a second phase, a consensus-reaching process is applied to

obtain group recommendations. Group recommendations are typically elicited out of individual personal recommendations through the application of mechanisms such as aggregation [12]. On top of that, our approach provides group members an additional feedback option by ranking the candidate group recommendations (Section `refgrouprec`). This facilitates the final selection of a travel destination for groups.

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 [25].

Another example in the domain of POIs is Group Modeller, a group recommender that provides information about museums and exhibits for small groups of people [18]. This recommender system creates group models from a set of individual user models.

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 [17]. 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 [16] 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.

An alternative recommender system for planning a vacation is CATS (Collaborative Advisory Travel System) [26]. 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.

Traditional recommender solutions often suffer from several shortcomings. Collaborative filtering solutions may suffer from the cold start problem to generate recommendations for new users that have not yet consumed any item, or for items that have not been consumed yet. Content-based recommendation

algorithms often suffer from over-specialization, which refers to the problem that recommended items are too similar to one another, thus reducing the diversity of results and limiting user choices [38]. To overcome these shortcomings, several recommendation algorithms have been combined into hybrid solutions [21]. Hybrid approaches, similar to our approach, have been proposed in the domain of tourism recommendations [28]. To recommend tourist attractions in Macedonia, a content-based component has been combined with collaborative filtering. For the content-based component, an explicit user profile based on tourist types [14] is used. This user profile is represented by a vector, which is continuously updated when a user gives a rating to an attraction. The cosine similarity is used to find the best match between the user profile and the tourist attraction features. For the collaborative filtering component, the algorithm uses a cloud model [36].

To address the challenges of the tourism domain, the TAST model (Tourist-Area-Season-Topic) proposes to represent the content as travel packages [20]. In this model, travel packages and tourists are represented by different topic distributions, where the topics are linked to the intrinsic features of the visited destinations. Their recommendation approach is to mix content-based (mapping all users and items to the TAST-model) and collaborative filtering (finding similar-minded users for the correct target area). In subsequent research [19], this model has been extended with a relation factor to a TRAST model, to also represent the relationship between travelers, making this model suitable for group recommendations.

In contrast to existing systems, the goal of our recommender system 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 intent to travel together. Moreover, users have a feedback option to evaluate intermediate group recommendations, thereby providing the group the opportunity to negotiate the travel options.

The remainder of this paper is structured as follows. Section 3 provides an overview of the architecture of the travel recommender system and the flow of data through it. 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 made. Section 8 gives the results of a user evaluation of the recommender systems. Finally, Section 9 draws conclusions from our research.

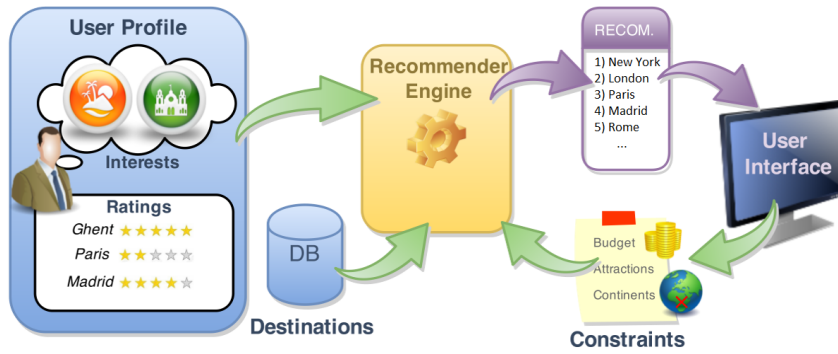


Figure 1 Overview of the data flow in the recommender service.

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 proper functioning of a recommender system depends on the availability of consistent, correct, and comprehensive data sources [6]. Specific personal preferences and user constraints, which are characteristic for the domain of traveling, emphasize the importance of data quality. This section provides an overview of the data that is used and the information sources.

The items, processed and outputted by the recommender, are all cities known for their tourism value. Many online services for POIs are available such

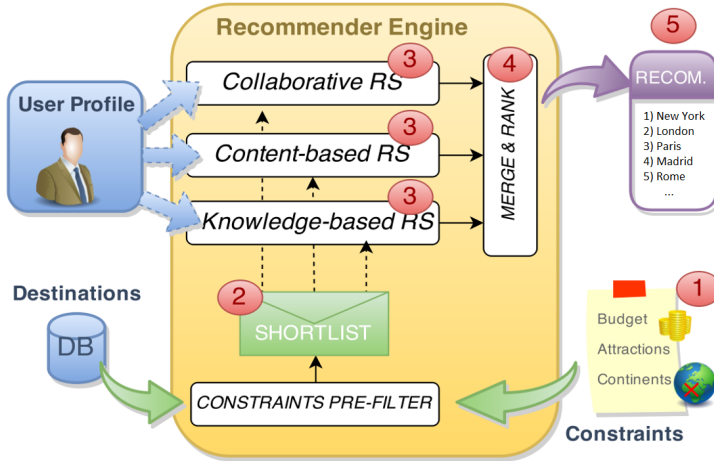


Figure 2 The system architecture of the recommender system.

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 dump of WikiVoyage [37], the Wikipedia alternative for travel destinations, which is available under the license by the Wikimedia Foundation. This information service consists of more than 26,000 locations and tourist information pages, created by users. One of the main advantages of this service, is that all entries have specific tourism value and come with a description and information 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 [13] 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 small, unimportant, unknown locations, which may be interesting to explore while in the neighborhood, but that have insufficient tourism value to be the main reason of a travel destination. Since these small 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 [35], 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 stored:

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

In order to obtain a typical tourist profile for each destination, the website Gogobot [15] is consulted. Gogobot is a travel application website that lets their users rate travel destinations and attractions. In comparison with other social travel networks such as TripAdvisor, Gogobot is unique by making use of *tribes*. Tribes represent different tourist profiles, which allow users to identify them with one or more of these 19 groups (e.g., Backpackers, Family Travelers, Adventure Travelers, Business Travelers, or Budget Travelers). 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 star rating from the user may be more suitable to the tribes of that users, than to other 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 individual web services allow to retrieve real-time prices for trains, airplanes, or another means of transport. The webservice Rome2rio [34], which was used in our travel service, combines different transportation methods and allows to retrieve transportation options and their predicted cost from any location in the world to any destination. 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 make the trip to 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 collected. Ratings for attractions were summarized to ratings for the destination where the attraction can be found. Ratings for destinations that are not in the destination database (or filtered out because of their tourism value), are not used and ignored in the calculations. Finally, 53,028 ratings were remaining and used by the recommender engine.

5 Travel web application

The travel 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 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 visible 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 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)

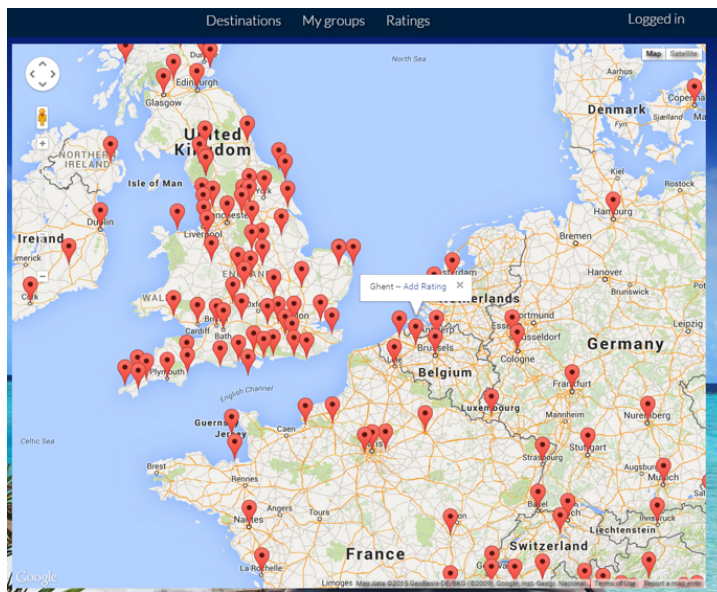


Figure 3 Screenshot of the user interface showing the possibility to navigate to a destination using Google Maps.

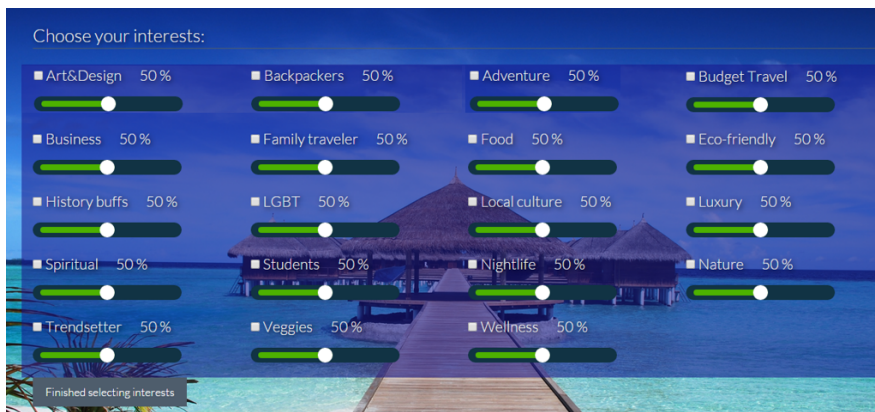


Figure 4 Screenshot of the user interface showing the possibility to explicitly specify preferences.

received five recommendation lists to compare the different algorithms: 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 to the recommendations.

Before applying the recommendation algorithms, a pre-filtering of the candidate destinations is performed. Destinations that do not fulfill the user constraints as well as the destinations already rated by the user, do not appear

on the shortlist. So, recommending already visited destinations is assumed undesirable. The resulting shortlist of destinations is used as input for the recommendation algorithms.

6.1 Collaborative filtering

As collaborating filtering algorithm, we opted for an item-item approach and used the implementation of the Lenskit Framework [10]. As input, a combination of explicit ratings and implicit feedback is used, given by users on Gogobot.com [15]. 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).

Besides, for each destination or attraction at a destination, users can indicate on Gogobot if they have “Been here”. This data is used as implicit feedback for the destination in the travel recommender since 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 an unseen 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 if two items are associated, but still favors destinations with a huge amount of feedback. 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 [9] is used for items that received an explicit rating from the target user in order to make a rating prediction \hat{r}^e .

$$\hat{r}_{u,i}^e = \bar{r}_i + \frac{\sum_{j \in \mathcal{N}_u^e(i)} \text{sim}(i,j)(r_{u,j} - \bar{r}_j)}{\sum_{j \in \mathcal{N}_u^e(i)} |\text{sim}(i,j)|} \quad (2)$$

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

$$\hat{r}_{u,i}^i = \frac{\sum_{j \in \mathcal{N}_u^i(i)} \text{sim}(i,j) * \bar{r}_j}{\sum_{j \in \mathcal{N}_u^i(i)} |\text{sim}(i,j)|} \quad (3)$$

Here, the neighborhood $\mathcal{N}_u^i(i)$ stands for the items j for which the target user u 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 [39].

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

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 much 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 as unreliable. In this case, recommendations using collaborative filtering might not be accurate enough given the small neighborhood size and the recommendations are disregarded. Then, the recommender system falls back on the content-based and knowledge-based approaches.

6.2 Content-based recommender

The idea of content-based recommendations 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 this travel recommender another approach, unique 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. The travel recommender utilizes the tags of attractions described on TripAdvisor [35], 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 node. To illustrate our approach we give an example for ‘Paris’ as 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) [22]. 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. To explain this with the above example of destination Paris for instance: 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 renowned 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 weights for the term frequency in literature [22]. 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 important 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 approach makes use of deeper connections and information provided by domain experts for the recommendations. 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). Hard constraints are handled by the pre-filter, which eliminates all destinations that do not fulfill any of these before the destinations are processed by the recommendation algorithms. The soft constraints are handled by the knowledge-based recommender, which gives a penalty to the destinations that do not completely fulfill the requirements. However, if the soft constraints are

not met, the destination can still end up in the final recommendation list if it matches 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 thereby 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.): how each location matches typical tourist profiles as defined in Gogobot [15].

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.

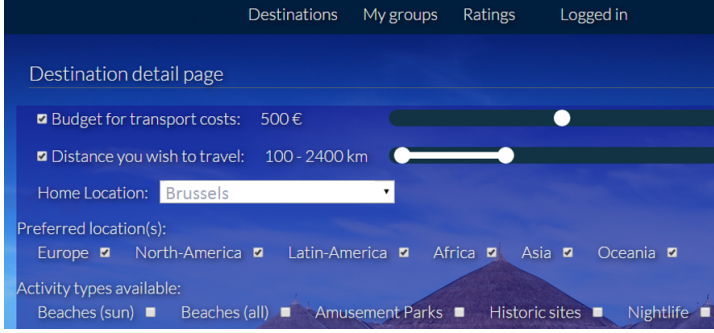
These constraints and user requirements are matched against the candidate destinations, providing a score for each dimension (location, costs, attractions, profile). Table 1 shows the scoring function for each dimension, as well as a weight for the relative contribution of each dimension to the rating 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 profile (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). As alternative, 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 [15]. 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

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

Dimension	Scoring $sc(i,k)$	w_k
Geo location	$1 - \sqrt{\frac{\max(\text{distance}(\text{user}, \text{location}) - \max_distance, 0)}{\max_distance}}$	1
Travel costs	$1 - \frac{\max(\text{expected_cost} - \max_budget, 0)}{\max_budget}$	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.

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.

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

Here, the summation is limited to the dimensions \mathcal{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 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 the results together combines the different information sources. In other words, the hybrid recommender should be a more robust recommender by making up for misjudgments of the individual recommenders.

Different approaches for generating hybrid recommendations are commonly used in literature, such as weighted models, switching models, mixed models, a

cascade of recommenders, and models that combine features of algorithms [4]. Because of the complexity of the domain of travel recommendations and the importance of user constraints, a model was chosen in which the information is merged at the end of the recommendation chain, such as the weighted, switched or mixed model. This way, each recommendation algorithm can fully exploit the available knowledge of user preferences and destinations. Since choosing a travel destination is an important decision for most people, a very robust hybridization method was chosen, namely the weighted model. With this hybridization strategy, destinations have a higher probability to be in the final recommendation list if they are recommended by more than one algorithm, and therefore come with a lower risk of being inappropriate.

To merge the rating predictions, a simple weighted sum of all three predicted scores is calculated. The different indices are cf for collaborative filtering, cb for the content-based recommender, and kb for the knowledge-based recommender.

$$\hat{r}_{hybrid} = w_{cf} * \hat{r}_{cf} + w_{cb} * \hat{r}_{cb} + w_{kb} * \hat{r}_{kb} \quad (7)$$

$$w_{cf} + w_{cb} + w_{kb} = 1 \quad (8)$$

These weights are not static, but determined based on 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 less (soft) constraints are specified by the user). Since an initial profile is created for each user, the content-based recommender has in general sufficient information to generate recommendations and can therefore act as the fall-back algorithm when both other approaches show little confidence. Alternative strategies to determine the weights are possible, such as adaptive weights depending on the accuracy of the individual recommendation algorithms. However, such a strategy is not adopted because of the rather long learning period to determine the optimal weights in comparison with the frequency of choosing a destination (typically a few times a year).

7 Group recommendation

Many travel plans are not made by individuals but by groups of people: friends, families, sport teams, etc. Besides individual recommendations, the travel recommender 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 available options. 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 [7]. 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 [7]. The reason for this is that aggregated recommendations for a group can be linked to recommendations for individuals, and as a result, can be justified more clearly in terms of an individual's preferences and constraints.

To aggregate the individual recommendations into group recommendations, various strategies are possible [24]. 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 [24]. 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 really disliked by one 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 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 to the group.

In the second phase, group members can give feedback on the list and choose 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 [24]. 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.

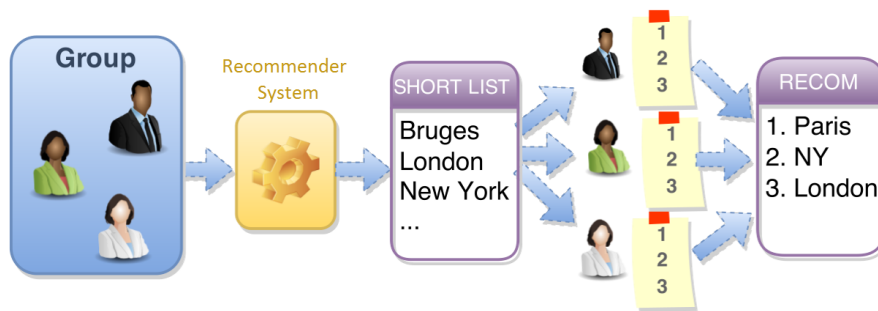


Figure 6 Schematic overview of the generation of group recommendations.

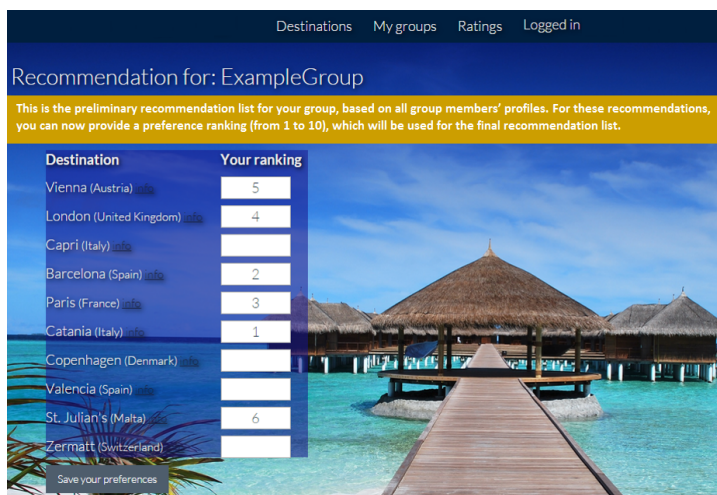


Figure 7 Screenshot of the user interface showing the possibility to rank the group recommendations.

8 Evaluation

The travel recommender application was presented to 16 users, who are representative for the target market of a travel service. For the evaluation, people are selected who travel regularly (once or several times a year) abroad and choose the destination themselves or at least have a say in the choice of the destination (group selection). Since the users are residents of Belgium, their holiday destinations are often located in Europe. These users have a variety of travel preferences such as cultural visits, nightlife activities, or exploring nature.

The users were asked to experiment with the recommender system and evaluate the different recommendation lists (collaborative filtering, content-based, knowledge-based, and hybrid recommendations, as well as a static list of popular destinations). The evaluation was performed in three phases.

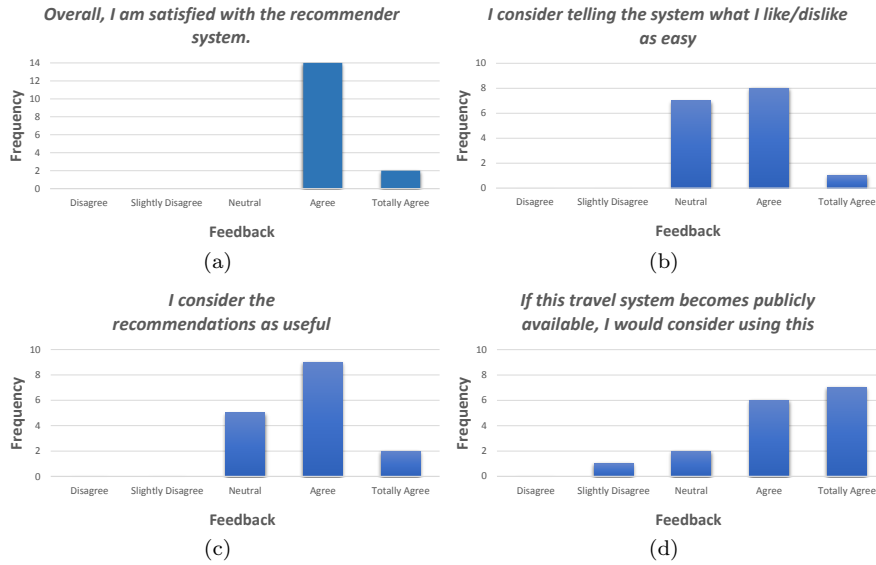


Figure 8 The results of the user evaluation regarding the general quality of the travel recommender system.

8.1 Travel recommender system

The goal of the first phase was to assess the general quality of the travel recommender system. For collecting some qualitative feedback regarding the service, each user was asked to fill in a questionnaire that is based on the evaluation framework of Pu [29]. 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 destinations they did not know before. They enjoyed the experience of finding 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)). Some 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)).

8.2 Comparison of the algorithms

The goal of the second phase was to assess the users' opinion about the quality of each recommendation algorithm: collaborative filtering (CF), the content-based recommender (CB), 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 (Pop). 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, (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.

To evaluate different quality attributes of the different algorithms, users were asked to evaluate the five recommendation lists using four multiple-choice questions.

1. Q1. [**Match interests**] The destinations recommended to me match my interests.
2. Q2. [**Novelty**] The recommender system helped me discover new destinations to visit.
3. Q3. [**Diversity**] The destinations recommended to me are diverse.
4. Q4. [**Usefulness**] The recommender gave me useful suggestions.

Users had to pick one of the five answers: disagree, slightly agree, neutral, agree, or totally agree, which are for evaluation purposes converted to 1, 2, 3, 4 and 5. Figure 9 shows the mean value of the obtained scores for each quality attribute.

These results show that the recommendations from the hybrid and the knowledge-based recommender are best *matching the users' preferences*. The accuracy of the knowledge-based recommender can be explained by the optimal match between the recommendations and the explicitly-stated user requirements and constraints. The hybrid recommender obtained a high accuracy by combining the data and knowledge of multiple individual recommenders. Compared to the hybrid recommender, the mean value of the perceived accuracy is about 0.5 lower for collaborative filtering and the content-based recommender. As expected, using the most-popular destinations as recommendations yields the lowest accuracy.

In terms of *novelty* of the recommendation, the best results are obtained by the collaborative filter. By using community knowledge, collaborative fil-

tering introduces serendipity into the recommendations, thereby providing novel and interesting suggestions. Recommendations of the content-based and knowledge-based recommender score slightly lower in terms of novelty, since these recommendations match the user’s constraints and requirements, and are often similar to previously visited destinations. The hybrid recommender is a combination of the other three algorithms, thereby balancing out some unexpected recommendations. These unexpected recommendations might be rather risky (suggested by only one of the three recommenders) and are therefore not adopted by the hybrid recommender. As a result, the hybrid recommender ends up with some more obvious choices, and the user might already be familiar with some of the destinations in the hybrid recommendations. Therefore, the hybrid recommender does not achieve the same level of novelty than the collaborative filter. Although novelty is a desired characteristic of recommendations, it may not be a goal on its own. Generating some random destinations to visit, for example, might end up with very novel recommendations, but with a very low accuracy (match to personal interests). Since the list of most-popular recommendations contains many classic, well-known destinations, this list has the lowest score on novelty.

Regarding the *diversity* of the recommendation list, the best score is obtained by the knowledge-based recommender. These recommendations can cover very different destination types, with different categories and local attractions, as long as the user requirements are met. By combining recommendations of three different algorithms, the hybrid recommender obtains a very diverse set of recommendations as well. Also the collaborative filter recommends a very diverse set of recommendations, because these are derived from the destinations visited by many different neighboring users. In contrast, destinations in the content-based recommendation list are often very similar in terms of category, local attractions, etc., resulting in a lower recommendation diversity. Also the list of most-popular destinations, containing all famous tourist cities, is typically characterized by a low diversity.

In terms of the *usefulness* of the recommendations as perceived by the users, the knowledge-based, hybrid, and content-based recommender achieved about the same score. Personal preferences or requirements, can be recognized in each of these recommendation lists. For recommendations based on collaborative filtering, it is often more difficult for users to see a direct connection between the personal preferences and the recommendations. As a result, these recommendations are perceived as a little less useful in comparison with the other algorithms. The most-popular recommendations are perceived as the least useful list. These recommendations are not personalized and typically well-known travel destinations.

Subsequently, the test users were asked to rank these five lists based on their own assessment of the suitability of the recommendations. This ranking allows users to evaluate how they really perceive the value of the recommendations. In other words, this question assesses the users’ experience with each recommendation algorithm. Whereas Figure 9 evaluates the algorithms in terms of individual quality metrics, Figure 10 shows the global assessment

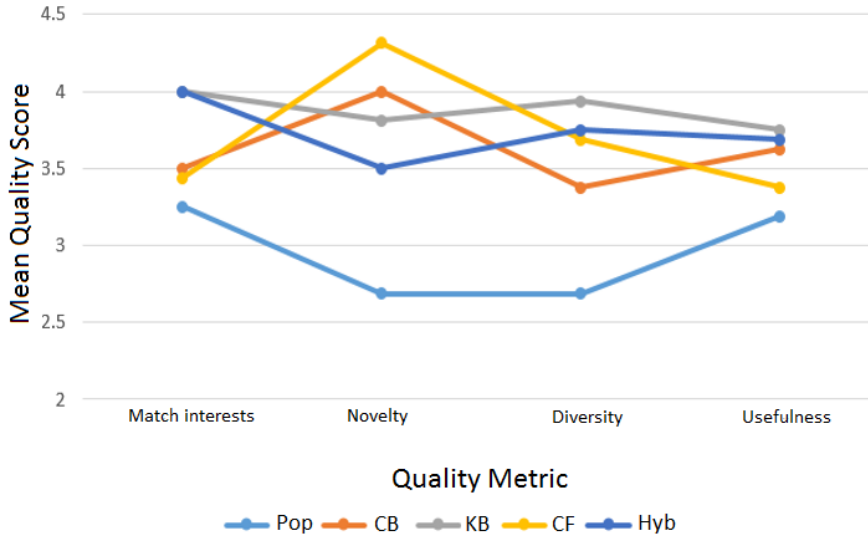


Figure 9 The mean values of the results scores of the five algorithms in terms of different quality attributes.

of the algorithms as experienced by the test users. Since the goal of this travel recommender system is to assist users in the choice of their destination as much as possible, their experience with the system can be considered as the most important evaluation metric.

Figure 10 shows the distribution of the obtained rankings for each algorithm. Rank 1 is assigned to the best algorithm, rank 5 denotes the worst 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 most-popular approach achieved the worst results (as expected). We assume 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 (Pop). The mean of the rankings assign 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 the results. The t-tests showed that the difference with the baseline recommender (Pop) in terms of mean ranking was statistically significant for the hybrid recommender ($p\text{-value} = 0.004$), the content-based recommender ($p\text{-value} = 0.028$), and the knowledge-based approach ($p\text{-value} = 0.031$). Only the collaborative filter did not show statistical evidence ($p\text{-value} = 0.251$) of receiving a better ranking than the baseline.

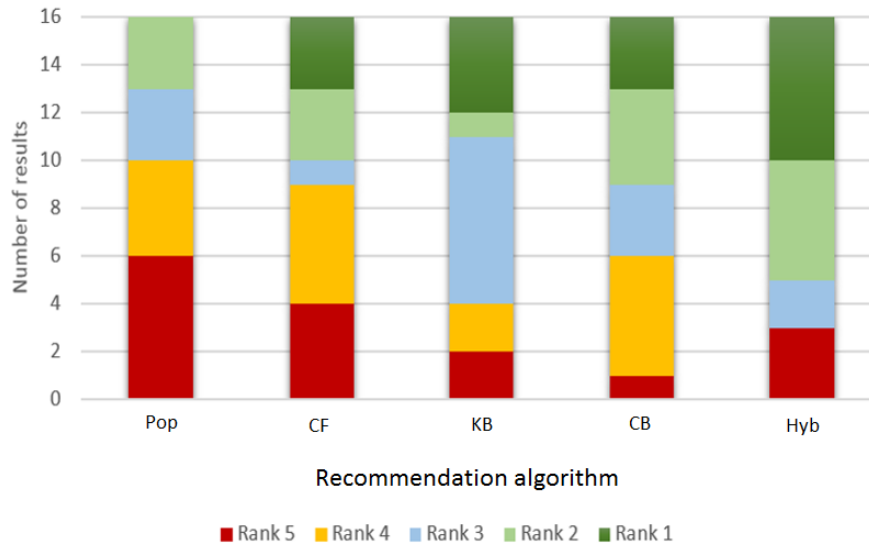


Figure 10 Distribution of the rankings in terms of user experience, given to the recommendation algorithms by the test users. Rank 1 is the best; Rank 5 is the worst.

8.3 Group recommendations

Besides individual recommendations, also group recommendations were offered to the test users, which are evaluated in the third phase. The 16 test users constituted four groups. Two groups could be described as a *family* group, with parents and siblings put together. One of these groups consisted of four family members, the other group had three members. The two remaining groups were groups of *friends*, sized six and three people respectively. As explained in Section 7, users were offered ten candidate recommendations for their group, and were subsequently asked to rank these according to their level of satisfaction. Afterward, they received a new list of group recommendations, adjusted to the group members' feedback. Finally, test users were asked to evaluate these group recommendations.

Figure 11 shows the results of the evaluation of the group recommendations per question. Most users, but not all of them, understand the group recommendations (Figure 11(a)). Since these group recommendations combine the preferences of all group members, the reason to recommend some of the destinations to the group is unclear for some users. In future versions of the service, explanations can help to explain the group recommendations in terms of individual group members' preferences.

The group recommender system takes the personal opinion of each user sufficiently into account (Figure 11(b)). The two phases of the group recommendation process help to match user preferences and destinations for the group. According to most users, there is a fair balance for all group members

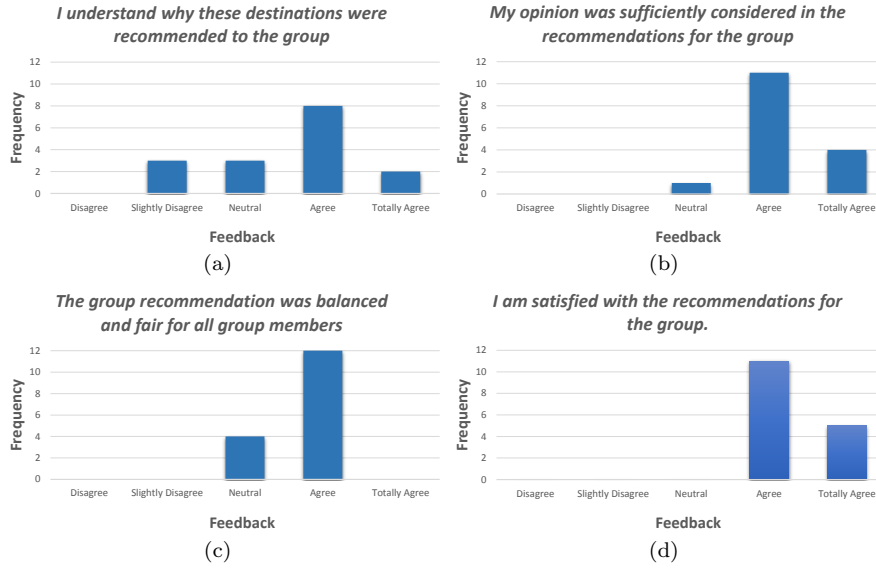


Figure 11 The results of the user evaluation regarding the quality of the group recommendations.

in the recommendation list (Figure 11(c)). So, every members' preferences are represented in the group recommendations about equally by using 'average without misery' as aggregation strategy. Finally, a question about the general perception of the group recommendations (Figure 11(d)) revealed that all users are satisfied with their group recommendations.

9 Conclusions

Because travel destinations proved to be a complex domain for recommendations, characterized by personal preferences, user constraints, and the typical group activity, no single algorithm would be able to consider all aspects for traveling. Moreover, gathering metadata and user feedback (ratings) showed to be not as trivial for travel destinations as 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. Comparison of different recommendation algorithms indicated that users prefer the hybrid recommendations in terms of overall experience 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 for the users. The group recommendation in two phases (aggregating recommendations for individuals + user feedback on the initial group recommendation list) received a positive evaluation in terms of understanding user preferences, taking into account and balancing user preferences, and overall satisfaction. 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 tour, or a region to explore.

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

References

1. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: Tailoring the recommendation of tourist information to heterogeneous user groups. In: S. Reich, M. Tzagarakis, P. De Bra (eds.) *Hypermedia: Openness, Structural Awareness, and Adaptivity, Lecture Notes in Computer Science*, vol. 2266, pp. 228–231. Springer Berlin Heidelberg (2002). URL http://dx.doi.org/10.1007/3-540-45844-1_26
2. Bollen, D., Knijnenburg, B.P., Willemsen, M.C., Graus, M.: Understanding choice overload in recommender systems. In: *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10*, pp. 63–70. ACM, New York, NY, USA (2010). DOI 10.1145/1864708.1864724. URL <http://doi.acm.org/10.1145/1864708.1864724>
3. Buhalis, D., Law, R.: Progress in information technology and tourism management: 20 years on and 10 years after the internet the state of tourism research. *Tourism Management* **29**(4), 609 – 623 (2008). DOI <http://dx.doi.org/10.1016/j.tourman.2008.01.005>. URL <http://www.sciencedirect.com/science/article/pii/S0261517708000162>
4. Burke, R.: Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* **12**(4), 331–370 (2002). DOI 10.1023/A:1021240730564. URL <http://dx.doi.org/10.1023/A:1021240730564>
5. Burke, R.: Hybrid web recommender systems. In: *The adaptive web*, pp. 377–408. Springer (2007)
6. De Pessemier, T., Dooms, S., Deryckere, T., Martens, L.: Time dependency of data quality for collaborative filtering algorithms. In: *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10*, pp. 281–284. ACM, New York, NY, USA (2010). DOI 10.1145/1864708.1864767. URL <http://doi.acm.org/10.1145/1864708.1864767>
7. De Pessemier, T., Dooms, S., Martens, L.: Comparison of group recommendation algorithms. *Multimedia Tools and Applications* **72**(3), 2497–2541 (2014). DOI 10.1007/s11042-013-1563-0. URL <http://dx.doi.org/10.1007/s11042-013-1563-0>
8. De Pessemier, T., Dooms, S., Martens, L.: Context-aware recommendations through context and activity recognition in a mobile environment. *Multimedia Tools and Applications* **72**(3), 2925–2948 (2014). DOI 10.1007/s11042-013-1582-x. URL <http://dx.doi.org/10.1007/s11042-013-1582-x>
9. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (eds.) *Recommender Systems Handbook*, pp. 107–144. Springer US (2011). DOI 10.1007/978-0-387-85820-3_4. URL http://dx.doi.org/10.1007/978-0-387-85820-3_4
10. Ekstrand, M.D., Ludwig, M., Kolb, J., Riedl, J.T.: Lenskit: A modular recommender framework. In: *Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11*, pp. 349–350. ACM, New York, NY, USA (2011). DOI 10.1145/2043932.2044001. URL <http://doi.acm.org/10.1145/2043932.2044001>
11. Felfernig, A., Burke, R.: Constraint-based recommender systems: technologies and research issues. In: *Proceedings of the 10th international conference on Electronic*

- commerce, ICEC '08, pp. 3:1–3:10. ACM, New York, NY, USA (2008). DOI 10.1145/1409540.1409544. URL <http://doi.acm.org/10.1145/1409540.1409544>
12. Garcia, I., Sebastia, L., Onaindia, E.: On the design of individual and group recommender systems for tourism. *Expert Systems with Applications* **38**(6), 7683 – 7692 (2011). DOI <http://dx.doi.org/10.1016/j.eswa.2010.12.143>. URL <http://www.sciencedirect.com/science/article/pii/S095741741001506X>
 13. GeoNames: The geonames geographical database (2015). Available at <http://www.geonames.org/>
 14. Gibson, H., Yiannakis, A.: Tourist roles: Needs and the lifecourse. *Annals of Tourism Research* **29**(2), 358 – 383 (2002). DOI [http://dx.doi.org/10.1016/S0160-7383\(01\)00037-8](http://dx.doi.org/10.1016/S0160-7383(01)00037-8). URL <http://www.sciencedirect.com/science/article/pii/S0160738301000378>
 15. Gogobot Inc: The go-to place for places to go (2015). Available at <http://Gogobot.com/>
 16. Jameson, A.: More than the sum of its members: challenges for group recommender systems. In: *Proceedings of the working conference on Advanced visual interfaces, AVI '04*, pp. 48–54. ACM, New York, NY, USA (2004). URL <http://doi.acm.org/10.1145/989863.989869>
 17. Jameson, A., Baldes, S., Kleinbauer, T.: Two methods for enhancing mutual awareness in a group recommender system. In: *Proceedings of the working conference on Advanced visual interfaces, AVI '04*, pp. 447–449. ACM, New York, NY, USA (2004). URL <http://doi.acm.org/10.1145/989863.989948>
 18. Kay, J., Niu, W.: 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). URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.90.4976>
 19. Liu, Q., Chen, E., Xiong, H., Ge, Y., Li, Z., Wu, X.: A cocktail approach for travel package recommendation. *Knowledge and Data Engineering, IEEE Transactions on* **26**(2), 278–293 (2014). DOI 10.1109/TKDE.2012.233
 20. Liu, Q., Ge, Y., Li, Z., Chen, E., Xiong, H.: Personalized travel package recommendation. In: *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 407–416 (2011). DOI 10.1109/ICDM.2011.118
 21. Lucas, J.P., Luz, N., Moreno, M.N., Anacleto, R., Figueiredo, A.A., Martins, C.: A hybrid recommendation approach for a tourism system. *Expert Systems with Applications* **40**(9), 3532 – 3550 (2013). DOI <http://dx.doi.org/10.1016/j.eswa.2012.12.061>. URL <http://www.sciencedirect.com/science/article/pii/S0957417412013024>
 22. Manning, C.D., Raghavan, P., Schütze, H., et al.: *Introduction to information retrieval*, vol. 1. Cambridge university press Cambridge (2008)
 23. Martinez, L., Rodriguez, R., Espinilla, M.: Reja: A georeferenced hybrid recommender system for restaurants. In: *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on*, vol. 3, pp. 187–190 (2009). DOI 10.1109/WI-IAT.2009.259
 24. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. In: *Personalized Digital Television, Human-Computer Interaction Series*, vol. 6, pp. 93–141. Springer Netherlands (2004). DOI 10.1007/1-4020-2164-X_5. URL http://dx.doi.org/10.1007/1-4020-2164-X_5
 25. McCarthy, J.: 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)
 26. McCarthy, K., Salamo, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P.: Cats: A synchronous approach to collaborative group recommendation. In: G. Sutcliffe, R. Goebel (eds.) *FLAIRS Conference*, pp. 86–91. AAAI Press (2006). URL <http://dblp.uni-trier.de/db/conf/flairs/flairs2006.html#McCarthySCMSN06>
 27. Moreno, A., Sebastián, L., Vansteenwegen, P.: Tours'15: Workshop on tourism recommender systems. In: *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15*, pp. 355–356. ACM, New York, NY, USA (2015). DOI 10.1145/2792838.2798713. URL <http://doi.acm.org/10.1145/2792838.2798713>
 28. Petrevska, B., Koceski, S.: Tourism recommendation system: empirical investigation. *Revista de turism-studii si cercetari in turism* (14), 11–18 (2012)

29. Pu, P., Chen, L., Hu, R.: A user-centric evaluation framework for recommender systems. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, pp. 157–164. ACM, New York, NY, USA (2011). DOI 10.1145/2043932.2043962. URL <http://doi.acm.org/10.1145/2043932.2043962>
30. Resnick, P., Varian, H.R.: Recommender systems. *Commun. ACM* **40**(3), 56–58 (1997). DOI 10.1145/245108.245121. URL <http://doi.acm.org/10.1145/245108.245121>
31. Ricci, F.: Travel recommender systems. *IEEE Intelligent Systems* **17**(6), 55–57 (2002)
32. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: *Recommender Systems Handbook*, 1st edn. Springer-Verlag New York, Inc., New York, NY, USA (2010)
33. Ricci, F., Werthner, H.: Case base querying for travel planning recommendation. *Information Technology & Tourism* **4**(3-4), 215–226 (2001)
34. Rome2rio: Rome2rio api and whitelabel (2015). Available at <http://www.rome2rio.com/documentation>
35. TripAdvisor Inc: Tripadvisor website (2015). Available at <http://www.tripadvisor.com/>
36. Wang, S., Xie, Y., Fang, M.: A collaborative filtering recommendation algorithm based on item and cloud model. *Wuhan University Journal of Natural Sciences* **16**(1), 16–20 (2011). DOI 10.1007/s11859-011-0704-4. URL <http://dx.doi.org/10.1007/s11859-011-0704-4>
37. Wikivoyage Inc: Wikivoyage the free worldwide travel guide that anyone can edit (2015). Available at <http://dumps.wikimedia.org/enwikivoyage/>
38. Yu, C., Lakshmanan, L., Amer-Yahia, S.: Recommendation diversification using explanations. In: Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on, pp. 1299–1302 (2009). DOI 10.1109/ICDE.2009.225
39. Zanker, M., Jessenitschnig, M.: Collaborative feature-combination recommender exploiting explicit and implicit user feedback. In: Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on, pp. 49–56 (2009). DOI 10.1109/CEC.2009.84