Mobile Dynamic Social Travel Recommender System

Abstract

Travel Recommender Systems (TRSs) help tourists discovering and selecting the Points of Interest (POIs) that best fit their preferences. Recommendations relay in the data available about the POIs of a destination, the knowledge about tourists and their preferences, and recommendation algorithms. This paper presents a dynamic social travel recommender system. The recommendation process is divided in two independent processes: the generation of user models and the calculation of the recommended POIs. The recommender generates user models and finds the similarities between users taking into account their explicit preferences, demographic information, rated POIs and tags created by users. These models are dynamically updated when new information is available. Then, a hybrid filtering algorithm combines these models with a content-based and a collaborative filtering algorithm to calculate a list of recommended POIs. The recommender has been integrated in a mobile prototype and preliminary results of its partial validation are presented.

Keywords: Travel Recommender System, user model, hybrid recommendation algorithm, mobile.

1 Introduction

The overwhelming amount of information about a destination and its Points of Interest (POIs) that is available makes the selection of the attractions or POIs to visit a difficult problem for tourists. Travel Recommender Systems (TRSs) assist tourists in this process, applying recommendation algorithms to estimate the POIs that best suit the preferences and needs of tourists. A TRS requires both detailed information about the POIs of the destination and tourists. The more information available, the more accurate would be the generated recommendations.

This paper describes the CRUMBS Travel Recommender System, which generates dynamic user models taking into account both implicit and explicit data and preferences of tourists, including their preferences about existing POIs. These models are applied into a hybrid recommendation algorithm to propose interesting POIs to tourists. The recommender is dynamic and social because the generation and categorization of new POIs, tags and ratings about any POI influence the behaviour of the recommender. Thus, both user models and recommendations are updated while tourists interact with the TRS.

This paper is organized as follows. First, Section 2 presents a summary of the state of the art. Section 3 introduces the dynamic social recommender system, describing the user modelling and recommendation algorithms. Section 4 presents the prototype integrating the recommender and the results of the partial validation of the system. The final Section presents some conclusion and future work.

2 State of the art

This Section presents the main research areas related to this paper: user modelling and Travel Recommender Systems. Interested readers are directed to review references for further information.

2.1 User modelling

The term "user model" can be used to describe a wide variety of knowledge about people (Rich, 1983). The information about a user, including his/her preferences is usually stored in a personal data structure known as profile. Three important dimensions that characterize user models have been identified by Rich (1979):

- One model of a single, canonical user which is necessarily uncertain but can represent users who have not usually interacted with the system vs. a collection of models of individual users.
- Models specified explicitly vs. models inferred by the system on the basis of the behaviour of the user.
- Long-term user models which represent demographics or general interests of the user vs. short-term user models that are suitable for a specific session or task.

The user model presented in this paper enhances a previous work based on stereotypes (Torre *et al.*, 2012) and considers the three dimensions. Firstly, it proposes a dynamic canonical user model based on the information available about tourists (demographic information, explicit preferences, tags and ratings). Secondly, models are based both on explicit data about users and information inferred from the tags and ratings they generate interacting with the system. Thirdly, user models are updated to build long-term user models taking into account the interaction of the user with the system.

Researchers have categorized examples of user modelling systems for tourism. Kabassi (2010) proposes a categorization of tourism user modelling systems based on the method of information acquisition: explicit, implicit or both of them. Moreno *et al.* (2012) categorize existing user modelling systems for tourism personalized recommender systems in three categories: demographic information, context-aware information and personal preferences.

2.2 Travel Recommender Systems

Travel Recommender Systems (TRSs) have been applied to recommend several tourism elements: from destinations to hotels, flights or POIs to visits when at the destination (Ricci & Werthner, 2002; Fesenmaier et al., 2003). TRSs can be categorized in three main groups according to the algorithm they apply: content-based filtering, collaborative filtering or hybrid approach (Kabassi, 2010). Traditional content-based filtering suggests items or services to a user, which are similar to those he/she bought or searched in the past by matching the characteristics of the item or service with the characteristics of the user that are maintained in his/her user model. In collaborative filtering approaches, recommendations are made by matching a user to other users that have similar interests and preferences. In this way, each user is suggested with items or services that other users, similar to the one interacting with the system, have bought before. Finally, hybrid approaches combine content-based and collaborative filtering methods in order to exploit their advantages and reduce their deficiencies (Adomavicius &Tuzhilin, 2005; Ricci *et al.*, 2011).

Mobile TRSs increase the potential and research challenges of traditional TRSs (Ricci, 2011). Recent examples propose the use of tags (Mikic-Fonte *et al.*, 2013) and user ratings to improve the quality of the recommendation (Yang and Hwang, 2013). Gavalas *et al.* (2013) presents a detailed survey of mobile TRSs, proposing a classification based on the architecture style, degree of user-involvement, and criteria deriving recommendations. Following such classification, the CRUMBS mobile TRS presented in this paper can be classified as a Web-based, pull-based, user constraints-based TRS.

3 Mobile Dynamic Social Recommender

3.1 General architecture

The CRUMBS TRS is based on a client-server architecture. The mobile client calls to the recommendation service and presents a list of recommended POIs on the mobile application. The server stores all the information and executes the user modelling and recommendation algorithms (Fig. 1). The recommendation service is published as a REST service returning a list of recommended POIs.

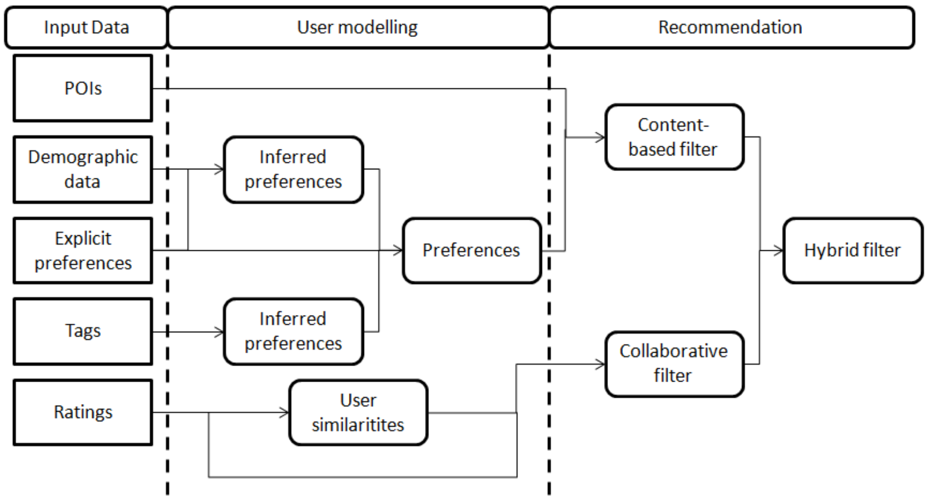


Fig. 1. General architecture of the CRUMBS Dynamic Social Recommender

Regarding the Input Data required by the Recommender, each POI can be related to different categories with different weights ranging between 0 (not related at all) and 100 (fully related). Basic demographic information (age, gender...) and explicit preferences of tourists are also taken into account. Explicit preferences refer to those preferences consciously expressed by the tourist in a scale ranging from 0 to 100, with zero representing displeasure and 100 representing the best score. Moreover, tourists add tags to POIs describing their opinion about them and rate values between 1 (most negative value) and 5 (most positive value) to POIs.

The recommender flow is divided in two phases. The first one is the user modelling to generate individual user models or profiles to predict the preferences and behaviour of tourists. Two types of inferred preferences are calculated with the same range as explicit ones (from 0 to 100): one from the tags created by tourists and the other combining the demographic information with the explicit preferences. Then, the inferred and the explicit preferences are combined to predict the preferences of each tourist. Ratings are also applied to find similarities between tourists on the assumption that people giving a similar rating to the same POIs share similar tastes.

On the other hand, the recommender applies a hybrid filtering algorithm to recommend POIs to tourists. This algorithm combines a content-based and a collaborative filtering algorithm. The former takes into account both the preferences of the tourists and the categories related to the POIs. The latter is based on the ratings and the similarities between tourists.

3.2 User modelling

3.2.1 Dynamic calculation of user preferences

User preferences about a certain category of POIs are calculated combining those explicitly expressed by the user and the ones inferred by the system. Given one specific preference p about a category c, the following equation adjusts its weights from different sources for each user.

$$p = \begin{cases} 0.7 * p_e + 0.3 * p_{i_T} & \textit{if the user has explicit preference} \\ 0.4 * p_{i_D} + 0.6 * p_{i_T} & \textit{otherwise} \end{cases} \quad (1)$$

where p_e is the value of the preference given explicitly by the user; p_{i_T} refers to the weight of the specific preference inferred from the tagging information; and p_{i_D} refers to the weight of the specific preference inferred from the demographic data of the user.

Regarding the calculation of the user preferences inferred from the tagging information, each time a tourist creates a tag about a POI, it is assumed that the tourist is interested on that POI and the categories related to it. Thus, the inferred preference p_{i_T} of a tourist in a category c can be calculated as the average weights of all the tagged POIs in that category:

$$p_{iT} = \frac{\sum^{P} w_{pc}}{|T_u|} * 100 \quad (2)$$

where Tu is the vector of all the tags created by a user u; P is the vector of the POIs related to the tags created by the user u; and w_{pc} is the weight of the relation between a POI p from P and a category c.

Furthermore, the calculation of the preferences of the user inferred from demographic information is based on a model-based approach to take into account demographics about tourists in order to extract some patterns to help assigning preferences to new tourists. The data considered are the age, the gender, the relationship status, the city of residence and the nationality of the tourist.

In order to ease the acquisition of their demographic information, tourists can import their profile information from different social networks (Facebook, Google+ and Twitter) once they have given the required permission on a social network.

When a new user has no explicit preferences, his demographic information allows extracting some patterns that could help assigning preferences. First, model-based algorithms are applied to generate user models. For each category the demographic data of tourists with a explicit preference about the category are selected as learning instances for the model. For each category, three models are generated using a different algorithm (linear regression multi-variable, model tree and rules). Preliminary tests have shown that the selection of the best algorithm strongly depends on the characteristics of the learning instances. Thus, the model obtained with each algorithm is cross-validated and only the model with the better Mean Absolute Error (MAE) is stored. Then, for each category the stored model is applied to predict the preference about the category for the user with no explicit preferences.

3.2.2 Similarities between tourists

The similarities between tourists are calculated from the POIs they rate applying a collaborative filtering algorithm (Algo. 1). For each tourist, similarity with other tourists who have co-rated at least one POI is computed using Pearson's correlation. Pearson's correlation corresponds to the cosine of users' deviation from the mean rating. Thus, it addresses variances in user ratings styles. For example, with a rating scale between 1 and 5, some tourists may give a rating of 5 to a liked POI and a rating of 3 to a disliked one, whereas another tourist with the same preferences will rate 4 a liked POI and 1 a disliked one.

As the similarity between two tourists is commutative and the similarity between one user and himself/herself does not provide interesting information, more than half of the values of the matrix are dismissed. Thus, the result of the collaborative algorithm is a triangular non invertible matrix with the similarity values among tourists S_{NxN}. Moreover, and to get a better space efficiency, the update process of the matrix only considers the *k* most similar users for each active user as neighbours. These selected neighbours are stored on the database and used later in the computation of prediction for each active user within the recommendation phase.

In order to establish the size of the neighbourhood, preliminary tests have been conducted to measure the relation between MAE and the size *k* of the neighbourhood to make a compromise between *k* and the calculation time. Tests have shown that increasing the size of the neighbourhood only leads to slight improvements on MAE. The optimal value of *k* will depend on the rating data and the time available for calculation, but a neighbourhood size of 100 returned good quality results with different data sets.

Algo. 1. Calculation of user similarities

Data: U: vector of N users (tourists) with at least one rating;
P: vector of M POIs;
R_{MxN}: matrix of ratings of the M POIs given by the N users;
r^i_{pj}: Rating given by a user u_j about a POI p rated by user u_i;
ur_p: Ratings given by a user u_i about a POI p also rated by user u_j;
or_p: Ratings given by a user u_j about a POI p also rated by user u_i;
Result: S_{NxN}: matrix of the similarity values between users;

```
foreach u_i ∈ U do
    foreach POI p rated by u_i do
        foreach u_j ≠ u_i do
            if R_pj ≠ 0 then
                r^i_pj = R_pj;
    foreach u_j where j > i do
        foreach POI p rated by u_i do
            if r^i_pj ≠ 0 then
                ur_p = R_pi;
                or_p = R_pj;
        cor_ij = Pearson(ur, or);
        if cor_ij > 0 then
            update(S_ij, cor_ij);
```

3.3 CRUMBS Travel Recommender System

3.3.1 Collaborative filtering

On the basis of the available ratings and user similarities, the Resnick's algorithm (Resnick et al, 1994) is used to compute the prediction for a target item i (POI) and a target user a (tourist). Prediction p_{ai} is a numerical value expressing the predicted likeliness of a item i not rated by the active user a. This predicted value is within the same scale ([1,5]) as the opinion values provided by a.

$$p_{ai} = \bar{r}_a + \frac{\sum_{v \epsilon V} sim_{av}(r_{vi} - \bar{r}_v)}{\sum_{v \epsilon V} |sim_{av}|} \quad (3)$$

where V is the set of k similar neighbours that have rated i; r_{vi} is the rating of i for neighbour v; $\bar{r_a}$ and $\bar{r_v}$ are the average ratings over all rated items for a and v; respectively; and sim_{av} is the Pearson correlation between a and v.

3.3.2 Content-based filtering

The features associated to each POI are defined by the author of the POI when this is created. Then, the correlation between the categories of the POI and the preferences of the tourist are computed using the vector similarity (Salton and McGill, 1983). In the context of the CRUMBS TRS, tourists and POIs take the role of documents, preferences over categories of POIs take the role of words, and weights take the role of word frequencies. The correlation between a user a and item i is calculated by the cosine of the angle between the two vectors representing them:

$$\cos(a, i) = \frac{\sum_j w_{aj} w_{ij}}{\sqrt{\sum_{k \epsilon I_a} w_{ak}^2} \sqrt{\sum_{k \epsilon I_i} w_{ik}^2}} \quad (4)$$

where the summations over j are over the categories for which both user a and item i have a defined weight, represented by w_{aj} and w_{ij} accordingly. I_a and I_i represent the group of categories set for user a and item i respectively. The cosine similarity computes ranges from 0 to 1, since the weights set to each category are on the [0,100] scale. As all the ratings in the system are within the 1-5 scale, the predicted likeliness p_{ai} of item i for the active user a will be calculated using a linear distribution:

$$p_{ai} = 4 \cos(a, i) + 1 \quad (5)$$

After repeating the calculation for an active user over all the existing items, a ranked list of POIs is produced and returned as a recommendation.

3.3.3 Hybrid Strategy

The CRUMBS TRS is based on a hybrid strategy that combines the content-based and the collaborative filtering algorithms through a linear combination to achieve a more accurate filter than each method alone. Thus, a prediction is based on a weighted average of the separately computed predictions. If the prediction for a tourist a over a POI i obtained from the content-based filtering is pcb_{ai}, and that obtained from the collaborative filtering is pc_{ai}, the resulting prediction is:

$$p_{ai} = x * pcb_{ai} + (1 - x) * pc_{ai} \quad (6)$$

Both techniques have high significance for the generation of recommendations, so they have the same weight (*x=0.5*).

3.4 Implementation

The CRUMBS Dynamic Social Recommender has been implemented in Java using the Open Source Weka library (www.cs.waikato.ac.nz/ml/weka [Aug. 28, 2013]). Weka is a collection of machine learning algorithms for data mining tasks that contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

The recommendation service is available through a REST service that returns a list with the recommended POIs for a user. The recommender has three configuration parameters. Firstly, the maximum size of the list of recommendations has been set to 10 to avoid presenting too many options on the mobile application. Secondly, the size k of the neighbourhood has been set to 100. Thirdly, the recommendation algorithm can be switched from the hybrid one to just the content-based or collaborative one.

Preliminary tests with synthetic data showed that the calculation time grown exponentially with the amount of tourists, POIs, rates and tags. Thus, some optimization techniques were required in order to obtain recommendations in real-time (less than five seconds).

First, as the dynamic characteristic of the user modelling algorithm does not require strict real-time execution, the related processes have been scheduled to be executed offline at night. Results of the calculated preferences and the similarities between tourists are stored on the database. Then, the recommendation algorithm retrieves this information from the database, generating a list of recommended POIs in real-time.

In order to further reduce the response time, recommendations are also generated offline and stored on the database daily. When a tourist asks for a recommendation, it is retrieved immediately if it is available on the database. For new users, it is calculated in real-time with a response time of less than five seconds.

Finally, as the recommendation of POIs is highly related to the current location of tourists, POIs are filtered according to the position of the tourist obtained from the mobile device during the recommendation phase. Thus, the amount of information to be processed by the algorithms is reduced, speeding up the calculation process.

4 Validation

4.1 CRUMBS prototype

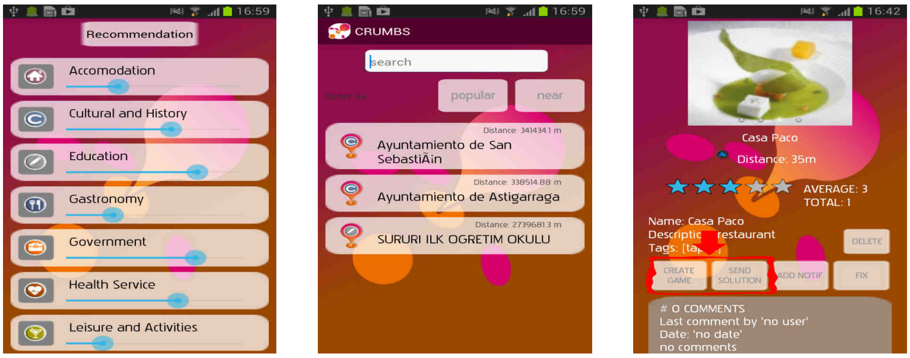
The mobile Dynamic Social Recommender has been integrated in the final prototype of the international CRUMBS project (crumbs.tid.es [Aug. 25, 2013]. CRUMBS proposes to re-organize information available in a social network based on the interaction of the users with their physical environments while wandering around. An entire geo-spatial social world has been created enabling the users to consume the multimedia social content "stuck" in different places as well as leave "crumbs" (POIs) as traces of their own activities. This is enabled by exploiting the functionalities of advanced Location-Based Services and Augmented Reality engine, mixing enriched social media content with the real world images captured by the camera of the mobile phone.

The CRUMBS prototype has been developed as an Android application and the recommendation functionality is integrated as a REST service. The offline processes have been included on the cron service of the CRUMBS server to be run daily at night.

Tourists enter their demographic information (Fig. 2a) either when they register on the system or using the configuration sections once they are logged in. They can also import their demographic information from Facebook, Twitter or Google+ (Fig. 2b).

(a) (b) (c)

Fig. 2. Mobile application. (a) Demographic information; (b) Authorizing to import the information from Facebook; (c) Main menu



(a) (b) (c)

Fig. 3. Recommendation service. (a) Explicit preferences of a tourist; (b) List of recommended POIs; (c) Details of a POI

Tourists have an option called “Interest” on the main menu (Fig. 2c). This option opens a screen presenting some sliders to adjust their explicit preferences (Fig. 3a). When tourists ask for the recommendation, the application presents the POIs on a list (Fig. 3b) with their name and the estimated distance. If tourists are interested on a POI, they can access further information about it (Fig. 3c).

4.2 Validation

The final validation of the CRUMBS TRS will be held with real users on November 2013 in San Sebastian together with the validation of the CRUMBS prototype. However, a preliminary validation has been performed in order to verify the recommendation service and its performance integrated on the CRUMBS prototype. Moreover, partial validations are being held by members of the consortium of the CRUMBS project before the final validation.

4.2.1 Preliminary validation

A preliminary validation has been performed with synthetic data to verify the recommendation service. The user modelling, content-based filtering and hybrid filtering algorithms have been tested with custom synthetic data, successfully validating their performance based on their computation time. As the accuracy of their predictions strongly depends on real data, this functionality will be validated during the final validation.

Regarding the collaborative filtering algorithm, the accuracy of its predictions has been validated with an offline experiment using a dataset provided by GroupLens Research. GroupLens has collected and made available rating datasets from the MovieLens web site with 100.000 ratings. The dataset has been divided into five parts (subfolders), each one containing 80% of the original data for the training phase and 20% of the original data for the test phase.

The evaluation of the prediction accuracy has been based on the performance measure called Mean Absolute Error (MAE). This metric computes the average of the absolute difference between the predictions and the true ratings. The process has been repeated for each of the subfolders for different neighbourhood sizes. Results have confirmed the accuracy of the predictions and have been used to set the size of the neighbourhood to 100.

Finally, the recommendation service has been integrated on the prototype and published as a REST service that has been tested within a user case with three actions: registering a user, sending his/her position and obtaining a recommendation. These actions have been reproduced automatically with a testing software generating close to 100.000 queries in 30 minutes with a delay between calls between 1 and 10 seconds. The response time for the worst case (500 users), have been below 4 seconds, which can be considered acceptable at this amount of load.

4.2.2 Partial validation

Partial validations are being held by members of the consortium of the CRUMBS project following a quality approach based on focus-groups. Two different groups have been identified: end-users (Focus Group Interviews) and expert workshops (Individual In-Depth Interviews).

The validations are based on 18 user cases (including recommendation) covering the demonstration of the main CRUMBS services and letting the interviewed use the application to catch their feedback on usability issues, as well as impressions on the functionalities of the application itself. These evaluations are still uncompleted and no conclusion can be obtained. However, first partial results show some interesting points related to the recommender system detected by a sample of users who are 34-40 years old and are familiarized with technology and social networks.

The recommendation of POIs has been perceived as one of the most positive functionalities of the prototype: users describe it as a very useful service for mobile applications, especially when travelling or discovering new places.

Users were concerned mainly with the level of intrusiveness of the application and the lack of control over the user modelling and recommendation algorithms of the application. Some users would not like their actions be monitorized and registered permanently, they would prefer to be able to control this process: activate/deactivate the learning process whenever they want, and select things that can be learned or not. Moreover, some users would also like to control the actions influencing the user modelling algorithm, avoiding a not wanted action leading to not useful recommendations (for example if they select something they are not interested in). Regarding the recommendation algorithm, some users would like to have advanced options to limit the maximum distance to the recommended POIs or to select the type of algorithm generating the recommendation.

Finally, the requirement of connectivity to obtain recommendations was perceived as a negative characteristic of the recommendation service, and mobile tourism applications in general. Most users would like to use the service when they are abroad, but they disable the data connection abroad due to the high cost of international roaming.

5 Conclusions

This paper presents a Mobile Dynamic Social Recommender system. The recommendation process relies on information about POIs and the categories they are related to, demographic information and explicit preferences of tourists about categories of the POIs, tags about POIs generated by tourist, and POIs rated by tourists. This is the input data to the user modelling and recommendation algorithms proposed by the system.

During the user modelling phase, preferences of tourists on the categories are predicted combining explicit preferences with two types of inferred preferences. First, the type of inferred preferences are calculated based on the tags created by tourists on the assumption that tagging a POIs suggests interest on the category related to it. Second type of inferred preferences are based on demographic information. For each category a user model is generated with the demographic information of tourist with explicit preferences about it. These models are applied to tourists with no explicit preferences to infer their preferences on the categories. Moreover, inside the user modelling phase the similarity between tourists is also calculated based on the assumption that tourists with similar ratings to the same POI have similar tastes. Then, a hybrid recommendation algorithm combines a content-based and a collaborative algorithm to generate a list of recommended POIs to a tourist.

The recommender has been integrated in the mobile prototype of the CRUMBS project, which proposes to re-organize information available in a social network based on the users interaction with their physical environments while wandering around. Although the final validation of the prototype is planned for November 2013, a preliminary validation has tested the performance of the recommender. Moreover, first results of partial validations have detected some key aspects for tourists when using the system.

Future work starts with the final validation, as it is required to measure the real relevance of the recommender and the acceptance of the system. However, results from the partial validations are already pointing out key lines for the future work.

Firstly, a solution giving tourists control of the information they share and the use of the information on the recommendation process is required to increase the trust of tourists in the recommender and the CRUMBS prototype. Secondly, integrating the recommendation engine with the mobile application in order to avoid the requirement of connectivity is an added value, mainly for scenarios where there are an important amount of foreign visitors.

References

Adomavicius, G. & Tuzhilin, A. (2005). Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowledge Data Eng*. 17(6): 734–749.

Fesenmaier, D. R., Ricci, F., Schaumlechner, E., Wöber, K. & Zanellai, C. (2003). DIETORECS: Travel advisory for multiple decision styles. In *Information and Communication Technologies in Tourism 2003*, 29–31, Helsinki, Finland. Springer.

Gavalas, D., Konstantopoulos, C., Mastakas, K. & Pantziou, G. (2013) Mobile recommender systems in tourism. *Journal of Network and Computer Applications*. Available online.

Kabassi, K. (2010). Personalizing recommendations for tourists. *Telematics and Informatics* 27(1): 51-66.

Mikic Fonte, F., López, M., Burguillo, J., Peleteiro, A. & Barragáns, A. (2013). A Tagging Recommender Service for Mobile Terminals. In *Information and Communication Technologies in Tourism 2013*: 424-435 Innsbruck, Austria. Springer.

Moreno, A., Valls, A., Isern, D., Marin, L. & Borràs, J (2012). SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities. *Engineering Applications of Artificial Intelligence* 26(1): 633-651.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. & Riedl, J (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work:* 175-186.

Ricci, F., & Werthner, H. (2002). Case based Reasoning for Travel Planning Recommendation. *Information Technology and Tourism*, 4(3-4):215-226.

Ricci, F., Rokach, L. & Shapira, B., eds (2011). Recommender Systems Handbook. Springer.

Ricci, F. (2011). Mobile Recommender Systems. *International Journal of Information Technology and Tourism* 12(3): 205-231.

Rich, E. (1979). User Modeling via Stereotypes. *Cognitive Science* 3(4): 329–354.

Rich, E. (1983). Users are individuals; individualizing user models. *International Journal of Man-Machine Studies* 18: 199-214.

Salton, G. & McGill, M. (1983). Introduction to Modern Information Retrieval. McGraw-Hill, New York.

Torre, I., Linaza, M-T. & Garcia, A. (2013). Hybrid User Modelling Algorithms for Tourism Providers. *In Information and Communication Technologies in Tourism 2013*: 483-494. Innsbruck, Austria. Springer.

Yang, W. & Hwang, S. (2013). iTravel: A recommender system in mobile peer-to-peer environment. *Journal of Systems and Software* 86(1): 12-20.

Acknowledgements

This paper is part of the CRUMBS project, contributed by 10 European organizations and co-funded by national resources under the Celtic-Plus scheme. The work presented in the paper has been financed within the Avanza program of the Ministry of Industry, Energy and Tourism of the Spanish Government (CRUMBS TSI-020400-2010-47).