

Context-Aware User Modeling Strategies for Journey Plan Recommendation

Victor Codina¹, Jose Mena¹, and Luis Oliva²

¹ Barcelona Digital Technology Center, Barcelona, Spain
`{vcodina,jmena}@bdigital.org`,
<http://www.bdigital.org>

² Technical University of Catalonia, Barcelona, Spain
`loliva@lsi.upc.edu`,
<http://www.lsi.upc.edu>

Abstract. Popular journey planning systems, like Google Maps or Yahoo! Maps, usually ignore user's preferences and context. This paper shows how we applied context-aware recommendation technologies in an existing journey planning mobile application to provide personalized and context-dependent recommendations to users. We describe two different strategies for context-aware user modeling in the journey planning domain. We present an extensive performance comparison of the proposed strategies by conducting a user-centric study in addition to a traditional offline evaluation method.

Keywords: Recommender systems, context-awareness, personalized journey planning, user-centric evaluation

1 Introduction

Nowadays, people commonly plan their holidays by making use of online tools that have access to a huge amount of tourist-related services and products, such as destinations, routes, attractions and hotels. This choice overload problem usually leads tourists to make poor decisions about what products are more appropriate to them.

Recommender Systems (RSs) [14][15] are known to be effective tools to overcome this problem, helping users to separate the wheat from the chaff. Previous approaches to journey plan recommendation are mostly designed for closed domains, that is, domains where the set of potential journey plans is already defined and based on a set of previously recommended Points of Interest (POIs) (*e.g.*, touristic itineraries) [10]; and those able to dynamically generate journey plans usually do not incorporate users preferences and context into the planning process. An exception is PECITAS [17], a mobile recommender that offers users personalized journey plans combining walking and public transportation in the city of Bolzano. However, in PECITAS only a small set of generic preferences are considered (*e.g.*, departure time and cost), which considerably limits its personalization capabilities.

Context-Aware Recommender Systems (CARSs) [1] differ from traditional recommendation strategies because they predict how a given user will rate an item not only based on past user ratings, but also exploiting the context in which those ratings were produced, and the user’s context at request time (*i.e.*, the target context). Previous research on CARSs in the tourism and travel domain has mainly focused on the task of POIs recommendation [2], not considering the journey planning itself as a recommending task.

Unlike previous works, we have investigated the effectiveness of applying state-of-the-art CARS techniques in an existing multi-modal journey planning system in order to provide users with personalized and context-dependent suggestions of dynamically generated journey plans. The main contributions of the paper are:

1. The definition of two different context-aware recommendation strategies for journey plan recommendation, one based on the adaptation of the contextual pre-filtering method presented in [5], and the other one a contextual modeling strategy inspired on the time-aware Matrix Factorization (MF) model presented in [8].
2. An extensive performance comparison of the proposed context-aware strategies combining offline and user-centric experiments. In the offline setting we used a contextually-tagged journey plan rating data set, collected by means of a novel in-context rating acquisition method, and for the user-centric evaluation we conducted a controlled experiment with real users of an existing journey planning mobile application.

The remainder of this paper is organized as follows. Section 2 positions our work with respect to the state of the art. Section 3 presents in detail the CARS strategies for journey plan recommendation evaluated in this work. Section 4 describes the offline evaluation, including the in-context rating acquisition method used to collect the training data. Section 5 presents the method and results of the user-centric experiment. Finally, Section 6 draws the main conclusions and presents some future work.

2 Related Work

CARSs are commonly classified into three paradigms [1]: (1) pre-filtering, where context is used for selecting the relevant ratings before computing predictions with a traditional context-free model; (2) contextual post-filtering, where context is used to adjust predictions generated by a context-free model; and (3) contextual modeling, in which contextual information is directly incorporated in a context-free model as additional parameters.

Most of the research on CARSs have focused on extending Collaborative Filtering approaches with context-awareness (CACF). Currently, pre-filtering and contextual modeling CACF methods extending MF prediction models are the most popular ones because of their superior prediction accuracy in many domains [8]. For example, *Distributional Semantic Pre-filtering* (DSPF) [5] is a

recent reduction-based pre-filtering method that, given a target context, builds a local MF prediction model using the ratings tagged with contexts semantically similar to the target one, which is then used for making predictions in that context. A key component of DSPF is the method used to compute the similarity between contextual situations based on the distributional semantics of their composing conditions. Another popular example following the contextual modeling paradigm is *Context-Aware Matrix Factorization* (CAMF) [2]. This method can be seen as a generalization of the standard bias MF model where context dimensions are included as part of the prediction function. Particularly, it incorporates context as additional biases capturing the global influence of contextual conditions on the ratings given to the items.

Pre-filtering and contextual modeling strategies have their own advantages and drawbacks, and it is difficult to know at design stage which algorithm would perform better in a specific recommendation domain [3][13]. A recent empirical analysis presented in [5] indicates that sophisticated pre-filtering methods can perform better in domains with high context granularity, but in contrast, contextual modeling strategies are less affected by data sparsity, since they use all the available training data to build the context-aware model. This uncertainty motivated us to carry out a more in-depth performance comparison of both context-aware strategies in the particular domain of journey planning.

Some authors have also investigated the effectiveness of extending Content-Based (CB) approaches with context-awareness (CACB). For instance, in [11] a contextual modeling method is proposed that learns attribute-based representations of both contextual information and user’s interests based on a vector space model enhanced by exploiting the attributes’ distributional semantics, which are then aggregated during the calculation of the user-item matching. However, a limitation of this approach is that it is designed to work with textual content data and, therefore, it cannot be applied in domains where only a limited set of categorical attributes are available like in this case. For this reason, differently from [11], in this work we adapted the previously mentioned CACF approaches, *DSPF* and *CAMF*, to be used in combination with a linear CB prediction model.

3 Context-Aware User Modeling Strategies

This section describes the context-aware recommendation strategies implemented to estimate the suitability of a candidate journey plan for a user, based on her past ratings and current context. We begin by describing the context-free recommendation model, which is the basis of the proposed CACB methods. Then, we present in detail the proposed context-aware strategies: the contextual pre-filtering method and the contextual modeling one.

3.1 Context-Free Recommendation Model

Collaborative Filtering (CF) approaches require several ratings of different users on the same items to extract meaningful interest patterns, which implies they

are not appropriate for recommending journey plans dynamically generated. In this scenario only CB approaches can produce meaningful recommendations, since they are capable to exploit the knowledge about the items besides the past ratings given by the target user.

To apply a CB approach we first need an attribute-based representation of the items to recommend (*i.e.*, journey plans). Based on the knowledge of mobility experts we identified as relevant the following set of attributes for the journey planning recommendation task:

- Global *time* required to complete the journey plan with respect to an estimated ideal time. In the case of multi-modal journey plans the time required for each transport mode (walk, bike, public and private transport) is also captured;
- *Cost* of the journey plan normalized with respect to the ideal cost. We also modeled the cost per mode of transport in multi-modal journey plans;
- *Physical effort* required (only available in plans including walking or biking as transport mode);
- *Time percentage* of each transport mode with respect to the global time;
- *Number of public interchanges* needed to complete the journey.

CB prediction models are typically designed to function with categorical data. Therefore, we first had to discretize the numeric attributes described above. To do this, we employed a fuzzy-set method [12] that assigns each possible value to one or two predefined categories. In particular, we divided each attribute into 5 equal intervals: *very low*, *low*, *normal*, *high* and *very high*. This method allows for a more accurate discretization by assigning a weight to the categories that are close to the boundaries separating two intervals.

Based on this weighted attribute-based representation, we defined a linear CB prediction model that estimates the utility of a given item i for a user u as the normalized dot product of the attribute-based user vector (p_u) and item vector (q_i):

$$\hat{r}_{ui} = |M_{ui}|^\alpha q_i^\top p_u \quad (1)$$

where: α is a normalization factor (hyper-parameter) that modifies the dot product score based on the total number of non-zero partial products, and whose value must be fine-tuned to the training data; p_u is the real vector representing the interest scores of user u in each possible attribute value (the model parameters to be learnt); and q_i is the real vector representing the attributes of item i (the vector dimensionality is equal to the number of possible attribute values).

Several methods can be applied to do this user profile learning process. After evaluating several strategies [4], in this work we decided to employ the stochastic gradient descent (SGD) method, widely used for learning the parameters of MF models [8], estimating the optimal values of the model parameters (p^*) by solving the regularized least squares optimization problem.

3.2 Contextual Pre-Filtering Strategy

Here we briefly describe the proposed contextual pre-filtering approach, which is an adaptation of DSPF, the reduction-based method presented in [5]. Given a set of training ratings and a target context, our adaptation of DSPF builds a local CB prediction model using the following two-step process:

1. **Ratings filtering.** Firstly, we select the relevant subset of ratings to make predictions in the given context. This subset contains the ratings tagged with exactly the target context and those whose context is similar enough to the target one. As in the original method, we consider as similar all the candidate contextual situations whose similarity to the target one is larger than a pre-defined threshold.
2. **Local model learning.** Then, a local recommendation model is learned using the relevant subset of in-context ratings selected in the previous step. Differently from the original DSPF method, where local MF models are built, here we learn the linear CB prediction model defined in Equation 1 instead.

To measure the similarity between contexts, we employed the same definition proposed by [5], which is based on the distributional semantics of their composing conditions, assuming that two contexts are similar if they influence ratings in a similar way. Particularly, we measured the influence of contextual conditions as the average rating deviation over the user’s ratings produced when the condition holds (i.e., the *per-user* perspective). For the same reason that CF approaches are not applicable in this domain the *per-item* perspective is also not useful in this application.

3.3 Contextual Modeling Strategy

The proposed contextual modeling strategy is inspired on CAMF [2] and the time-aware MF model presented in [8], which extends the bias MF with additional parameters capturing the temporal dynamics as well as periodic effects. Our approach consists in extending the linear CB prediction model defined in Equation 1 by capturing the influence of context at two different granularities:

- As global biases, which capture the global context-specific variability of users’ preferences with respect to the attributes of the target journey plan. Denoting as b_{i_a, c_j} the rating bias associated to the item value of the attribute i_a when condition c_j holds, the global bias of a target contextual situation $s = [c_1, c_2, \dots, c_k]$ (k represents here the k -th contextual factor captured by the system, and c_k is the specific condition of this k -th factor in the target context) is calculated as the bias summation of s composing conditions:

$$\hat{b}_{ui, c_1 \dots c_k} = \sum_{j=1}^k b_{i_a, c_j} \quad (2)$$

- As preference-specific biases, which capture the context-dependent variability of the target user’s interests. More formally, we estimate the user’s preference vector p_u as a function of the target context, as follows:

$$\hat{p}_{u,c_1\dots c_k} = p_u + \sum_{j=1}^k \sum_{l=1}^o p_{u,c_l} \cdot S_{c_j,c_l} \quad (3)$$

Where: $S_{c_1,c_2} \in [0,1]$ stands for the similarity value between a candidate condition c_1 and the target condition c_2 , and o represents the total number of contextual conditions captured by the system. This semantic aggregation of context-specific user’s preferences is aimed to enhance the contextual user modeling process under new-user cold-start scenarios. In this case we used the same context similarity definition as in DSPF (see Section 3.2).

Based on the previous definitions, the implemented contextual modeling strategy, called here *Distributional-Semantics Contextual Modeling* (DSCM), predicts the rating that a target user u and item i under context s by summing the estimations of Equations 2 and 3, resulting with the following estimation function:

$$\hat{r}_{ui,c_1\dots c_k} = \hat{b}_{ui,c_1\dots c_k} + |M_{ui}|^\alpha q_i^\top \hat{p}_{u,c_1\dots c_k} \quad (4)$$

As in the context-free recommendation model (Section 3.1), we employed the SGD method to learn the DSCM’s model parameters (b^* and p^*). However, as suggested by [7], given the higher complexity of DSCM’s model here we split the learning process in two sequential phases: firstly, we learn the global contextual biases and static user’s preferences independently, and then, we learn the preference-specific contextual biases fixing the global ones. An extended discussion of the implemented user profile learning process can be found in [4].

4 Offline Evaluation

4.1 In-Context Rating Data Collection

To evaluate the effectiveness of the proposed context-aware recommendation approaches in the journey planning domain from a system-centric perspective, we first had to collect a data set of in-context user ratings for a variety of journey plans. To this aim, and based on the methodology presented in [2], we designed a web-based application in which users could make journey plan requests under imaginary contexts and rate them. Figure 1 shows the user interface design of this application. On the top-left corner there are the fields to specify the origin and destination locations, and at the bottom there is the context selection panel where users can specify the current contextual situation.

Based on the knowledge of mobility experts we identified 10 contextual factors as relevant in this domain: 2 user-specific factors, *companionship* and *purpose of journey*, and 8 environmental-based factors (*time of day*, *weather*, *temperature*, *crowdedness*, *illumination*, *moisture*, *pollution* and *pollen concentrations*).

The selected conditions are shown by an icon that represents it and, disabled or *unknown* factors are represented by a question mark icon. For example, in Figure 1 there are 5 enabled factors, which describe the following contextual situation: traveling alone, sunny day, high pollution, and high pollen concentration. By default, a set of contextual conditions was randomly picked. Initially, 2 factors at most were enabled to simplify the process. This constraint was gradually relaxed after some provided ratings. Once selected the context and the destination point, the user could request a plan by clicking the *Plan Journey* button. In response, the journey planner generated a non-personalized list of journey plans each of them using a different transport mode combination. Users were asked to analyze and evaluate all the suggested plans. Each suggestion was accompanied by a detailed explanation of the proposed journey plan, and users were able to vote them using a 5-star scale. We also remarked users to rate the journey plans taking into consideration the "active" contextual situation.



Fig. 1. In-context rating acquisition user interface

This experiment lasted one week, and we collected a total of 3256 ratings given by 68 users to 1628 journey plans in 736 different in-context journey plan queries. The set of participants was composed by 40% women and 60% men living in Barcelona (Spain) at the time of the experiment. Each user provided 48 ratings on average.

4.2 Experimental Results

We measured the accuracy of the recommendation models in terms of rating prediction accuracy calculating the Root Mean Square Error (RMSE) on a test set built from the collected journey plan rating data set. To build the training and test sets we used a variant of the *all-but-n* [16] splitting method, where for each user we randomly selected a small portion of her ratings for testing. Based on the

design of our in-context rating acquisition application (see previous section), we did not select the ratings individually but in terms of requests (*i.e.*, considering all the ratings given under a particular request and context). For each user we selected the ratings of 2 requests at most, equivalent to 10 test ratings on average. We calculated the statistical significance of RMSE differences by means of the Wilcoxon sign rank test.

The experimental results demonstrate that both context-aware strategies, *DSPF* and *DSCM*, clearly outperform the context-free baseline, whose absolute error is 1.27. Particularly, *DSCM*'s RMSE is 1.09 (reducing by 14% the error with $p\text{-value}=0.002$) and *DSPF*'s RMSE is 1.15 (9% reduction with $p\text{-value}=0.008$). Based on these results it seems that *DSCM* outperforms *DSPF*; however, their per-user RMSE differences are not statistically significant at the 95% confidence level ($p\text{-value}=0.25$). Therefore, we were not able to validate that *DSCM* is the best performing strategy just with this offline study.

5 User-Centric Evaluation

5.1 Experiment Methodology

In order to validate the apparent superiority of *DSCM* shown in the offline study, we conducted a controlled experiment with a different set of users. They were asked to evaluate the quality of the context-dependent journey plan recommendations in real scenarios, that is, under real contexts while travelling around the city. To this aim, we integrated the proposed recommendation strategies into an existing journey planning mobile application, which implemented a sophisticated multi-modal planner based on modified Dijkstras and A* search algorithms on top of generalized time-dependent graphs [9].

Figure 2 illustrates some screenshots of the application's user interface, which allowed users to set the origin and destination points as well as the user-specific contextual factors; the environmental factors were captured from the available city sensors (Section 4.1 shows the complete list of contextual factors). Finally, the best journey plan candidates were presented to the mobile users as a ranking sorted by predicted user's rating.

The experiment was based on A/B testing, in which users are randomly split in groups using different configurations of the functionality being evaluated. Particularly, we divided participants in 3 groups, each using one of the approaches evaluated in the offline experiment: the context-free baseline, *DSPF* and *DSCM*. The experiment was divided in two phases of 1 week long each:

- **Phase 1.** The goal of this phase was two-fold: (1) to bootstrap the recommendation model by collecting an initial set of users' ratings; (2) to let users experiment with a version of the system with limited personalization capabilities. During this phase all the users received recommendations based only on their user transport preferences (explicitly provided during user's registration). In particular, given a candidate journey plan, a rating estimation was generated as the summation of interest scores in each journey's

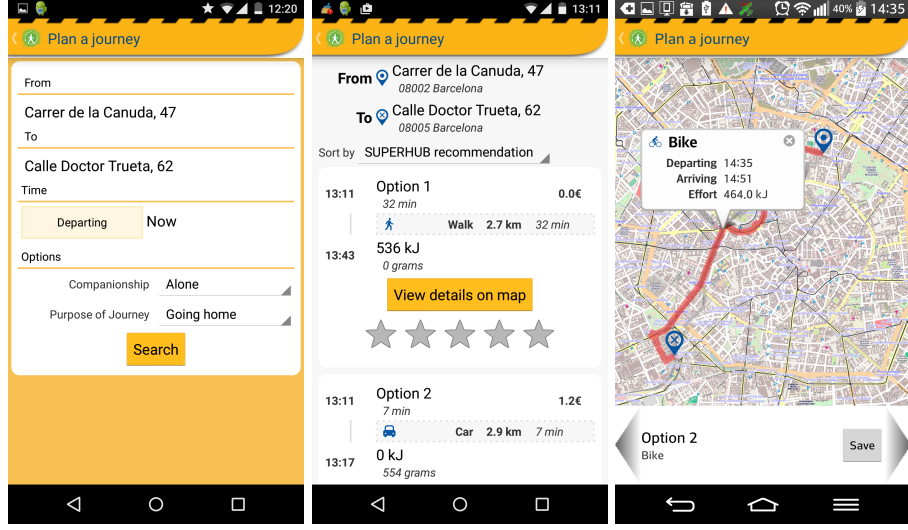


Fig. 2. In-context journey plan request (left), recommendation list (center) and map view (right) of the mobile app user interface

transport mode weighted according to the mode’s time percentage. Users were asked to rate the recommended rankings.

- **Phase 2.** We used the collected ratings during the first week for building the context-free and context-aware recommendation models as well as for hyper-parameter fine-tuning. At the beginning of this phase we communicated to the users that a new version of the mobile app, with more sophisticated context-aware recommendation functionality, was available. We also asked them to use the app again under at least six different contextual situations so users could fairly evaluate the effect of context-dependent recommendations. In this phase each user received recommendations generated by the prediction model corresponding to her group. Once the task was completed, users were asked to fill out a questionnaire to evaluate the quality of recommendations across 4 different aspects, as shown in Table 1.

Table 1. User study questionnaire across 4 different quality aspects

Top-n accuracy (TA)	The suggested routes fitted my preferences
Ranking accuracy (RA)	The suggested routes were ranked according to my preferences
Context-awareness (CA)	The suggested routes were adapted to my travel context
Overall satisfaction (OS)	The new recommendation service helped me to find better routes

5.2 Results and Discussion

Figure 3 shows the evaluation scores given by the 67 users that completed the experiment. Users were able to express agreement or disagreement with the 4 statements on a five-point scale (where 1=*strongly disagree*, 2=*disagree*, 3=*neither agree nor disagree*, 4=*agree*, and 5=*strongly agree*). Bar charts show the mean score value given by each users' group: *Free* (the context-free baseline), *DSPF*, and *DSCM*, to the four analyzed aspects. The error bar represents the 95% confidence interval. As significance test here we used the Welch's t-test, which is more appropriate for *between-user* studies.

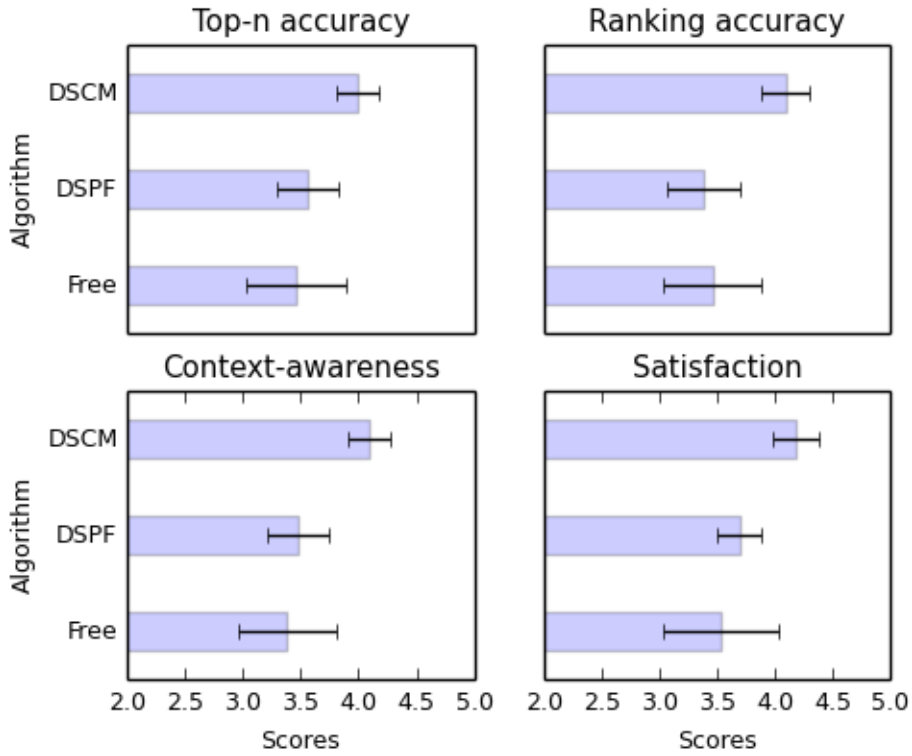


Fig. 3. User study results

Due to the limited amount of data in our study we followed the pragmatic user-centric evaluation framework presented in [6], where t-tests and simple correlations between the evaluated aspects are used to evaluate the effect of the proposed context-aware strategies on the user experience.

We firstly tested whether users of the three algorithms judge the recommendation accuracy differently. The results showed that the algorithm with the

highest mean response for the accuracy statements is *DSCM*. For example, in terms of RA, *DSCM* obtained a 4.1 mean score compared to the 3.4 and 3.5 scores of *DSPF* and *Free*, respectively. The t-tests showed that this superiority is statistically significant ($p=.005$ w.r.t. *DSPF* and $p=.05$ w.r.t. *Free*).

Then we tested whether users perceive algorithm’s CA differently. Comparing the context-aware strategies, one can observe that *DSCM* is again the algorithm with the highest score (4.1 versus 3.5, with $p=.004$).

Finally, we tested for a difference in terms of overall satisfaction. Again we can observe that *DSCM* is clearly the best evaluated algorithm w.r.t. *DSPF* (4.2 versus 3.7, with $p=.007$). Additionally, we analyzed how the subjective system aspects (TA, RA and CA) correlate with the overall satisfaction (OS). To this aim, we calculated the Pearson correlation between the scores given to each aspect. The results show that both aspects are strongly and significantly correlated to OS: .50 with TA, .51 with RA and .48 with CA.

Based on these results we can conclude that *DSCM* has a higher perceived recommendation accuracy and context-awareness than *DSPF*, which leads to a higher user’s satisfaction. Unexpectedly, in this study we did not find statistically significant differences between the results of *DSPF* and *Free*. We conjecture this poor performance of *DSPF* can be due to a bad hyper-parameter selection, since differently from *DSCM*, in *DSPF* the degree of contextualization strongly depends on the similarity threshold specified. To confirm this hypothesis we plan to carry out A/A testing for evaluating more precisely different *DSPF*’s configurations .

6 Conclusions and Future Work

This paper presents our experience in applying state-of-the-art context-aware recommendation technologies in a journey planning mobile application, which dynamically generates multi-modal journey plans when requested, in order to provide users with personalized and context-dependent journey plan suggestions. To this aim, we have developed and evaluated two different context-aware recommendation strategies: (1) a sophisticated reduction-based pre-filtering method that exploits finer-grained situation-to-situation similarities to build local content-based prediction models optimized for the target context; and (2) a contextual modeling method that extends a linear content-based model with additional parameters able to capture the context-variability of user’s preferences as well as global periodic effects. We have presented an extensive performance comparison of the proposed strategies combining offline with user-centric experiments, showing that the contextual modeling strategy is the best performing strategy.

In the current journey planning prototype, the context-aware recommendation strategies are applied at the end of the plan generation process for ranking. In future versions we want to investigate the effectiveness of the proposed strategies when applied during the construction of the plans. We conjecture that in this way the recommender can have a major impact and also help to speed up the plan generation process by incrementally reducing the search space.

Acknowledgments. This work has been supported by the FP7 IP Project SUPERHUB, funded by the European Commission (FP7-ICT-2011-7 ICT-2011.6.6, no. 289067). The authors want to specially acknowledge the Catalan Agency of Innovation and Internationalization (ACCIÓ) for their research support.

References

1. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-aware recommender systems. *AI Magazine* 32(3), pp. 67–80 (2011)
2. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing* 16(5), pp. 507–526 (2012)
3. Campos, PG., Fernández-Tobías, I., Cantador, I., Díez, F.: Context-Aware Movie Recommendations: An Empirical Comparison of Pre-Filtering, Post-Filtering and Contextual Modeling Approaches. In: *EC-Web’13*, pp. 137–149 (2013)
4. Codina, V., Oliva, L.: Contextual modeling content-based approaches for new-item recommendation. In: *IliX ’14*, pp. 259–262. ACM (2014)
5. Codina, V., Ricci, F., Ceccaroni, L.: Distributional semantic pre-filtering in context-aware recommender systems. *User Model User-Adapt. Interact.* doi:10.1145/2542668. Springer (2015)
6. Knijnenburg, Bart P and Willemsen, Martijn C and Kobsa, Alfred: A pragmatic procedure to support the user-centric evaluation of recommender systems. In *RecSys’11*, pp. 321–324 (2011)
7. Koenigstein, N., Dror, G., Koren, Y.: Yahoo! Music Recommendations: Modeling Music Ratings with Temporal Dynamics and Item Taxonomy. In: *RecSys ’11*, pp. 165–172. ACM (2011)
8. Koren, Y., Bell, R.: Advances in collaborative filtering. *Recommender Systems Handbook*, pp. 145–186. Springer (2011)
9. Hrnčir, J., Jakob, M.: Generalised Time-Dependent Graphs for Fully Multimodal Journey Planning. *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2138–2145 (2013)
10. Hsieh, Hsun-Ping and Li, Cheng-Te and Lin, Shou-De: Measuring and Recommending Time-Sensitive Routes from Location-Based Data. *ACM Transactions on Intelligent Systems and Technology* 5(3), pp. 1–27 (2014)
11. Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Combining Distributional Semantics and Entity Linking for Context-Aware Content-Based Recommendation. In: *UMAP’14. LNCS vol. 8538*, pp. 381–392 (2014)
12. Nguyen, T., Wu, B.: Random and Fuzzy Sets in Coarse Data Analysis. *Comput. Stat. Data Anal.* 51(1), pp. 70–85. Elsevier Science Publishers B. V.(2006)
13. Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., Pedone, A.: Comparing context-aware recommender systems in terms of accuracy and diversity. *User Model User-Adapt. Interact.* 24(1-2), pp. 35–65 (2014)
14. Ricci, F.: Travel recommender systems. *IEEE Intelligent Systems* 17(6), pp. 55–57 (2002)
15. Ricci, F., Rokach, L., Shapira, B.: Introduction to Recommender Systems. *Recommender Systems Handbook*, pp. 1–35. Springer (2011)
16. Shani, G., Gunawardana, A.: Evaluating Recommendation Systems. *Recommender Systems Handbook*, pp. 257–297. Springer (2011)
17. Tumas, G., Ricci, F.: Personalized Mobile City Transport Advisory System. *Information and Communication Technologies in Tourism*, pp. 173–183. Springer (2009)