# Context-Aware Places of Interest Recommendations and Explanations

Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci

Free University of Bozen-Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
lbaltrunas,bernd.ludwig,fricci@unibz.it

**Abstract.** Contextual knowledge has been traditionally used in Recommender Systems (RSs) to improve the recommendation accuracy of the core recommendation algorithm. Beyond this advantage, in this paper we argue that there is an additional benefit of context management; making more convincing recommendations because the system can use the contextual situation of the user to explain why an item has been recommended, i.e., the RS can pinpoint the relationships between the contextual situation and the recommended items to justify the suggestions. The results of a user study indicate that context management and this type of explanations increase the user satisfaction with the recommender system.

## 1 Introduction

Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user [8]. It is a matter of fact that more compelling and useful recommendations can be identified if the context of the user is known [1]. Here we adopt the definition of context provided by [5]: context is "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". For instance, in a travel recommender, the season and the duration of the travel are important contextual conditions that should be considered before suggesting a holiday.

For this reason, context-aware recommender systems (CARSs) have attracted a lot of attention, and in particular in the tourism domain [4, 3, 7]. But, in order to adapt the recommendations to the user's context one must first identify all the potential contextual factors that may influence the acceptance of a recommendation, e.g., distance to a target place of interest, motivations for the travel, etc. This knowledge can be obtained by referring to the vast consumer behavior literature, especially in tourism [9]. But this knowledge can only be used as a starting point. In a necessary second step the quantitative dependency of the user preferences (ratings for items) from each single contextual factor must be modeled. This dependency model can be built, in collaborative filtering RSs, by acquiring explicit ratings for some of the items to be recommended under several possible contextual conditions. So for instance, in our application domain, one should acquire the ratings of a museum (place of interest – POI) when the user is traveling with or without children or when she is alone.

In this paper, we briefly illustrate ReRex, a recommender system for places of interest (POIs), that exploits a context-aware rating prediction model to generate more useful recommendations and can explain the recommendations by referring to some selected factors describing the contextual situation of the target user [10]. We illustrate the evaluation methodology, based on the comparison of ReRex with a variant obtained by removing its context-awareness capability and recommendation explanations, showing that these two features of the system increase the user satisfaction with the recommender system.

## 2 Ratings in Context

Our working hypothesis is that a recommendation can be explained plausibly if at least the most important criteria that lead to the recommendation are communicated to the user. In our context-aware recommendation model, besides the user-item-matrix of ratings, the context, i.e., the set of conditions that hold when the recommendation is made, is of major importance for the recommendation.

Evidence that context matters for good recommendations is taken from a user study that we conducted. In this study subjects were asked to rate a selection of places of interest in Bolzano imagining that certain contextual conditions hold [2]. Table 1 lists some of the contextual factors that change the average ratings of particular categories of points of interest significantly (for lack of space only a selection of these categories is considered). For instance, "walking paths" are rated worse at "night time" or if the user is "far away" from that path. Note that in the table MCY, is the mean rating for items in that category when that contextual condition was considered, while MCN is the mean rating for the same selection of items when context was not considered.

This difference in the rating means is significant ($p < 0.001$: $***$; $0.001 \leq p < 0.01$: $**$; $0.01 \leq p < 0.05$: $*$). From this results we can conclude, for instance, that the rating prediction for a walking path should decrease if the user is far from it. Moreover, the distance to a walking path could be used as an argument for not suggesting that item even if based on other elements, e.g., the previous ratings of the user for similar items, it may seem a good recommendation. In contrast to this example, the mean rating of a walking path grows significantly if the user is with friends or she is in a lazy mood. Consequently, in that contextual conditions, the recommender could argue for its recommendation of a walking path by pointing out that since the user is with friends (or is in a lazy mood) then that particular walking path is a suitable activity.

The collected context-dependent ratings have been used to train a novel context-aware rating prediction model that extends and adapts the approach presented in [6]. We have introduced one model parameter for each contextual condition and item pair. To keep our approach tractable, we have modeled context as a set of independent contextual factors. The model then learns how the

**Table 1.** Effects of context on the mean rating for items. MCY is the mean of the ratings when that context is considered, while MCN is the mean of the ratings for the same items when context is not considered.

| **contextual condition** | **factor** | $p$**-value** | **MCN** | **MCY** | **Effect** |
|---|---|---|---|---|---|
| | | **Castle** | | | |
| far away | distance | $***$ | 3.80 | 2.47 | ↓ |
| winter | season | $**$ | 3.81 | 2.63 | ↓ |
| | | **Museum** | | | |
| sad | mood | $***$ | 2.79 | 1.64 | ↓ |
| activity/sport | travel-goal | $***$ | 2.64 | 1.33 | ↓ |
| active | mood | $***$ | 2.64 | 1.44 | ↓ |
| far away | distance | $**$ | 2.78 | 1.92 | ↓ |
| | | **Walking Path** | | | |
| night time | day-time | $***$ | 3.78 | 1 | ↓ |
| far away | distance | $***$ | 3.86 | 2.38 | ↓ |
| cold | temperature | $***$ | 3.8 | 1.88 | ↓ |
| winter | season | $***$ | 3.91 | 2.33 | ↓ |
| with friends or colleagues | companion | $***$ | 3.85 | 4.83 | ⇑ |
| crowded | crowdedness | $**$ | 3.88 | 2.75 | ↓ |
| working day | day-week | $**$ | 3.94 | 2.75 | ↓ |
| half day | time-available | $**$ | 4.01 | 1.6 | ↓ |
| more than a day | time-available | $**$ | 3.89 | 4.8 | ⇑ |
| lazy | mood | $**$ | 4.03 | 4.71 | ⇑ |

ratings deviate from classical personalized predictions as effect of one selected contextual factor, for each possible value of the factor, i.e., contextual condition. This deviation is the *baseline* for that contextual condition and item combination. Broadly speaking, the system computes a rating prediction for a user-item pair and then adapts that prediction to the current contextual situation, i.e., a combinations of contextual conditions (values for contextual factors) using the learned context-dependent baselines.

More precisely, in our data set of context-aware ratings, a rating $r_{uic_1\ldots c_k}$ indicates the evaluation of the user $u$ for the item $i$ made in the context $c_1, \ldots, c_k$, where $c_j = 0, 1, \ldots, z_j$, and $c_j = 0$ means that the j-th contextual factor is unknown, while the other index values refer to possible values for the j-th contextual factor. The tuples $(u, i, c_1, \ldots, c_k)$, for which rating the $r_{uic_1\ldots c_k}$ is known, are stored in the data set $R = \{(u, i, c_1, \ldots, c_k) | r_{uic_1\ldots c_k} \text{ is known}\}$. Note, that in our collected data set, only one contextual condition is known and all the others are unknown, hence in $R$ there are ratings for which only one among the indices $c_1, \ldots, c_k$ is different from 0.

The proposed model computes a personalized context-dependent rating estimation using the following equation:

$$\hat{r}_{uic_1 \ldots c_k} = \boldsymbol{v}_u \cdot \boldsymbol{q}_i + \bar{\imath} + b_u + \sum_{j=1}^{k} B_{ijc_j} \tag{1}$$

where $\boldsymbol{v}_u$ and $\boldsymbol{q}_i$ are $d$ dimensional real valued vectors representing the user $u$ and the item $i$. $\bar{\imath}$ is the mean of the item $i$ ratings in the data set $R$, $b_u$ is the baseline parameter for user $u$, and $B_{ijc_j}$ are the parameters modeling the interaction of the contextual conditions and the items. The parameters $\boldsymbol{v}_u$, $\boldsymbol{q}_i$, $b_u$, and $B_{ijc_j}$ are learned using stochastic gradient descent; this has been proved to be an efficient approach for similar learning problems [6].

In order to generate the explanation for a recommendation for item $i$ in the contextual situation $c_1 \ldots c_k$ we identified $j = \arg\max_j B_{ijc_j}$, i.e., the factor that in the predictive model has the largest positive effect on the rating prediction for item $i$. Using one single factor in the generated explanation has the benefit of creating a simple, easy to grasp motivation, and to not overload the user. The implementation of a concrete recommender system, which is using this model, is discussed in the next section.

## 3 The ReRex Mobile Application

In a typical interaction with ReRex the user initially establishes the context of the visit. Using the system GUI the user can enable and/or set the values of important contextual factors. The user can switch on/off some of these factors, e.g., the "Temperature" or "Weather" (see Figure 1, left). When one of these factors is switched on the recommender system will take into account its current value in the recommendation generation process. The full set of contextual factors considered in ReRex, their values (contextual conditions), and whether they are automatically collected, using an external service, or manually entered by the user, is provided in the following:

- Distance to POI (automatic): far away, near by;
- Temperature (automatic): hot, warm, cold;
- Weather (automatic): sunny, cloudy, clear sky, rainy, snowing;
- Season (automatic): spring, summer, autumn, winter;
- Companion (manual): alone, friends, family, partner, children;
- Time day (automatic): morning, afternoon, night;
- Weekday (automatic): working day, weekend;
- Crowdedness (manual): crowded, not crowded, empty;
- Familiarity (manual): new to city, returning visitor, citizen of the city;
- Mood (manual): happy, sad, active, lazy;
- Budget (manual): budget traveler, price for quality, high spender;
- Travel length (manual): half day, one day, more than a day;
- Means of transport (manual): car, bicycle, pedestrian, public transport;

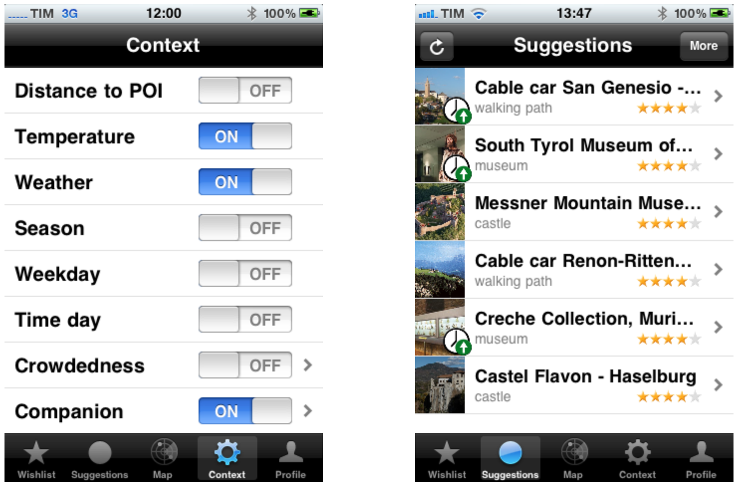


**Fig. 1.** ReRex context management (left); display for recommendations (right).

– Travel goal (manual): visiting friends, business, religion, health care, social event, education, cultural, scenic/landscape, hedonistic/fun, activity/sport.

After the user has entered the specification of the contextual situation (see Figure 1, left) the system can be requested to provide some recommendations. A short number of suggestions, namely six, are provided (see Figure 1, right). The recommendations are ordered according to their predicted rating. If the user is not happy with these suggestions she can request more recommendations. In the suggestion list the user can touch any of these suggestions to access a more detailed description of the POI (see Figure 2). It is worth noting that some of these suggestions are marked with an icon showing a small clock and a green arrow. This means that these recommendations are particularly suited for the current context of the request as it was previously acquired. For these recommendations (Figure 2) there is an explanation sentence like "This place is good to visit with family". This refers to the contextual condition that was largely responsible for predicting an high ranking for this item. Note, that "with family" condition could even decrease the rank of some items, i.e., their relevance for the current context. However, some items become more attractive than others (this specific museum in our case) if the group is a family. The other items, i.e., those not marked with the clock icon, are suited as well for the current contextual situation. But we decided not to explain their relationship with the context to highlight and better differentiate those marked with the clock icon from the rest. This can be considered as a persuasive usage of the contextual information.

We have identified custom explanation messages for all the possible 54 contextual conditions listed previously. We note that even if more than one contextual condition holds in the current recommendation session, and all of them are actually used in the computation of the predicted score of each recommenda-


**Fig. 2.** ReRex screen for explaining recommendations.

tion, nevertheless the system exploits only one of them for the explanation. The contextual condition that is used in the explanation is the most influential one as estimated by the predictive model used by the recommender to predict the relevance (rating) of items in the current context. This design choice is motivated by a simplicity reason; we hypothesized that a single statement would be easily understood by the users and ultimately would produce the best effect on them. Naturally this issue, and more in general a better explanation functionality could be implemented in a future version of the system. In fact, as it will be illustrated in the next section, the quality of these canned explanations were not perceived by the users as strikingly good, indicating that better explanation messages could be generated.

Some additional functions have been implemented to enable the user to better exploit the system. The user can add a recommendation to her wish list, rate an item, show the position of an item on the map. We also note that ReRex recommendations are updated when a relevant contextual condition is changed either by the user manually or is automatically acquired.

## 4 Experimental Evaluation

In order to measure the effectiveness of this approach we developed two variants of our ReRex mobile recommender system. The first one is that described previously, the second variant is not context-aware, i.e., there is no possibility for the user to specify the current context, the UI screen shown in Figure 1 (left), has been removed, and no recommendation is marked with any icon, or explained to stress the appropriateness for the current contextual situation. The prediction model described in Equation 1 is simplified in this second variant, and the parameters $B_{ijc_j}$ are not learned. This variant does not offer any explanation for the recommendation. Hence, comparing these two variants we could check if context management in the prediction model and the proposed explanation technique have a joint effect on user satisfaction compared to a system that does not exploit context at all.

To achieve this goal the test participants, 20 in total, tried out both variants of the system (within groups experimental model), in a random order, and executed, supported by each system, two similar but different tasks, related to travel planning. After the user completed the assigned task using one system, she was requested to fill out a usability questionnaire. These questions were extracted, and slightly adapted to the scope of our investigation, from the IBM Computer System Usability Questionnaire. Then finally the subjects were requested to compare the two systems. The full set of results of this evaluation are reported in detail elsewhere and are beyond the scope of this paper [2]. In summary, we can report that when the users were requested to directly compare the two variants, 85% of the users preferred the context-aware version, and 95% of the users considered the context-aware recommendations more appropriate. with respect to the explanation functionality, the subjects rated their agreements to the following two statements: (Q14) I am satisfied with the provided contextual explanations; and (Q15) I believe that the contextual explanations are useful. We observed a score of 1.05 for (Q14), and a higher score of 1.5 for (Q15) (scores range from -2, strongly disagree, to 2, strongly agree). This shows that the quality of the explanations is not yet optimal but the users clearly perceived the importance of such feature. Summarizing the evaluation results we observe that, even if this conclusion is supported by a limited number of testers, the context-aware recommendations were considered more effective than those produced by the non context-aware version. Moreover, the users largely agreed on the importance of explanations even if they complained about the quality of them. This indicates that the explanation is a very important component, it strongly influenced the system acceptance, but the user is particularly sensible to the quality of these explanation; and the formulation of these explanations can be surely improved.

## 5 Conclusions and Future Work

In this paper we have illustrated the importance of exploiting a traveler contextual conditions when recommending POIs. The proposed mobile application offers to the user context-aware recommendations that are justified and explained by referring explicitly to the contextual situation in which the user will experience them. We have shown that the proposed system can offer effective context-aware explanations that are generated by identifying the contextual conditions that show the largest influence on the predicted relevance score (rating) of the recommended items. In a live user study we have compared a context-aware version to a non context-aware one. We have shown that the user acceptance and satisfaction is larger for the context-aware version and that the users prefer this version compared to another, with a very similar user interface, which does not consider the request context and does not provide any explanations.

In a future work we want to better understand the individual role of personalization, contextualization, and explanations. In fact, in the study described in this paper we have compared a system offering contextualization of the recommendations and explanations with a variant that misses both features. We need to perform new experiments where the individual features are considered independently: for instance, comparing two context-aware systems: with and without explanations. A second issue was mentioned already in the paper and refers to the measured low user satisfaction for the generated explanations. We want to improve the quality of the explanations exploiting advanced natural language processing techniques to better adapt the explanation to the type of recommended item and using more information extracted from the predictive model.

## References


1. G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. Kantor, editors, *Recommender Systems Handbook*, pages 217–256. Springer Verlag, 2011.
2. L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. *(to appear) Personal and Ubiquitous Computing*, 2011.
3. V. Bellotti, J. Begole, E. Chi, N. Ducheneaut, J. Fang, E. Isaacs, T. King, M. Newman, K. Partridge, B. Price, P. Rasmussen, M. Roberts, D. Schiano, and A. Walendowski. Activity-based serendipitous recommendations with the magitti mobile leisure guide. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008*, pages 1157–1166. ACM Press, 2008.
4. F. Cena, L. Console, C. Gena, A. Goy, G. Levi, S. Modeo, and I. Torre. Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. *AI Communication*, 19(4):369–384, 2006.
5. A. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.
6. Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, KDD '09, pages 447–456, New York, NY, USA, 2009. ACM.
7. F. Ricci. Mobile recommender systems. *Journal of Information Technology and Tourism*, 12(3):205–231, 2011.
8. F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In F. Ricci, L. Rokach, B. Shapira, and P. Kantor, editors, *Recommender Systems Handbook*, pages 1–35. Springer Verlag, 2011.
9. J. Swarbrooke and S. Horner. *Consumer Behaviour in Tourism.* Butterworth-Heinemann, 2 edition, 2006.
10. N. Tintarev and J. Masthoff. Designing and evaluating explanations for recommender systems. In F. Ricci, L. Rokach, B. Shapira, and V. Kantor, editors, *Recommender Systems Handbook*, pages 479–510. Springer Verlag, 2011.