Intelligenza Artificiale 8 (2014) 129–143
DOI 10.3233/IA-140069
IOS Press

Techniques for cold-starting context-aware mobile recommender systems for tourism

Matthias Braunhofer*, Mehdi Elahi and Francesco Ricci
Free University of Bozen-Bolzano, Bozen-Bolzano, Italy

Abstract. Novel research works in recommender systems have illustrated the benefits of exploiting contextual information, such as the time and location of a suggested place of interest, in order to better predict the user ratings and produce more relevant recommendations. But, when deploying a context-aware system one must put in place techniques for operating in the cold-start phase, i.e., when no or few ratings are available for the items listed in the system catalogue and it is therefore hard to predict the missing ratings and compose relevant recommendations. This problem has not been directly tackled in previous research. Hence, in order to address it, we have designed and implemented several novel algorithmic components and interface elements in a fully operational points of interest (POI) mobile recommender system (STS). In particular, in this article we illustrate the benefits brought by using the user personality and active learning techniques. We have developed two extended versions of the matrix factorisation algorithm to identify what items the users could and should rate and to compose personalised recommendations. While context-aware recommender systems have been mostly evaluated offline, a testing scenario that suffers from many limitations, in our analysis we evaluate the proposed system in live user studies where the graphical user interface and the full interaction design play a major role. We have measured the system effectiveness in terms of several metrics such as: the quality and quantity of acquired ratings-in-context, the recommendation accuracy (MAE), the system precision, the perceived recommendation quality, the user choice satisfaction, and the system usability. The obtained results confirm that the proposed techniques can effectively overcome the identified cold-start problem.

Keywords: Cold start, recommender systems, context, mobile, personality, active learning

1. Introduction

In recent years, there has been an explosive growth of the sheer volume of the information available through the World Wide Web. For instance, the amount of travel offers, music, books, movies, images and web pages, accessible to people through the Web is continuously increasing, making it more and more difficult for any person to find out interesting and relevant items. For instance, users accessing tourism portals often find it extremely difficult to select a good hotel or a place to stay, due to the overwhelming number of offers and the lack of an effective system support.

*Corresponding author: Matthias Braunhofer, Free University of Bozen-Bolzano, Bozen-Bolzano, Italy. E-mail: mbraunhofer@unibz.it.

Recommender Systems (RSs) address this information overload problem by suggesting a small set of items that are judged to be interesting to the user [31]. These suggestions are typically computed by comparing the user's profile (which appropriately models the user preferences, tastes, and interests) with the description of the items (content-based approach) or with the profiles of other users (collaborative-filtering approach).

Context-Aware Recommender Systems (CARSs) are a special type of RSs: they aim at generating more relevant recommendations by adapting the recommendations not only to the user's preferences but also to the contextual situation [1]. For instance, in a tourist attraction recommender system, it is important to consider the weather when the recommended places will be visited. In fact, on bad weather conditions a tourist might prefer to visit indoor attractions (e.g., museums, churches, castles), while on good weather conditions, she might prefer to visit outdoor attractions (e.g., lakes, mountain lodges, scenic walks). In this case, the reasoning process is apparently simple, however, it can become more sophisticated and could also depend on the user's individual reaction to specific contextual conditions and on the interaction of different contextual factors (e.g., weather and time). For instance, in the system described in this article we use 14 contextual factors (see Section 3.4); in addition to the weather and the temperature, we consider also: the timing of the visit (season, daytime, weekday, duration), the location of the point of interest (distance from the user location, crowdedness of the place), the transportation mean of the user, her budget, mood, companion and travel goals.

Numerous research [2, 9, 16, 18, 20, 27] and commercial systems, such as Foursquare[1], Yelp[2], Pandora[3] have already been successfully implemented: all of them exploit the current user's contextual situation when recommending items. However, adequately addressing the so-called cold-start problem is still a challenge; this occurs typically when the system is initially deployed and no or just few ratings are available. We note that ratings for items are used by the majority of RSs to acquire user preferences, and in particular by collaborative filtering (CF) systems, which is the technology considered in this article. CF systems base their recommendations on the analysis of the target user's ratings and of the ratings of users with similar rating behaviours [31]. In a CF system a target user's unknown rating for an item, which the system must decide whether to recommend or not, is predicted by observing and averaging the ratings of similar users (neighbours). Recommendations are then generated by suggesting to the target user the items with the largest predicted ratings.

In order to deal with the cold-start, a CARS needs to collect an adequate set of rating data that are augmented with the information of the contextual situation of the user while experiencing the rated item. So, for instance, if the user u rated the item i 4 stars, this is stored in a two dimensional matrix $r_{ui} = 4$ of a non context-aware system. In a CARS this matrix is multidimensional: $r_{uic_1 \ldots c_k}$. The c_j indexes run over the possible (index) values of the contextual factors. This means, going back to example mentioned above, that in a CARS one must acquire also the information of what contextual conditions were observed when the user rated the item in order to assign $r_{uic} = 4$, which means that the user u rated the item i 4 when context was c (assuming here that there is only one contextual factor, e.g., the temperature). Therefore, in CARS the cold-start problem is even more severe: the space of possible ratings is larger and, for instance, without ratings collected in a particular contextual situation it is difficult to make relevant recommendations in that situation.

Hence, it is clear that it is not enough to have many ratings; they must be acquired in several alternative contextual conditions in order to let the system learn the users' contextually dependent preferences and to provide them with relevant recommendations for items under these various contextual situations. This is quite challenging given the high number of possible contextual situations observed in realistic scenarios. In our system we tackled this problem, as it will be better illustrated in the rest of this article, by requesting the user to rate certain items and by specifying only the most influential conditions, which were identified in a previous study where we analysed the correlations between contextual conditions and ratings. In order to identify the items to rate, we have designed and deployed a special strategy, which is based on the knowledge of the user personality. Using this approach we were able to collect more ratings, compared to another state of the art strategy, and we were able to improve more the recommendation quality.

1.1. Article scope and contributions

In summary, this article deals with the cold-start of CARSs. We have designed, applied and analysed a technical solution in a real mobile application that is called South Tyrol Suggests (STS) [6, 12]. STS is an Android-based RS that provides users with context-aware recommendations from a repository of approximately 27,000 POIs, including accommodations, restaurants, attractions, events and public services, that are located in the South Tyrol region of Italy.

In the early stages of its deployment, STS was facing an extreme cold-start situation, i.e., there were only a few hundred ratings, augmented with the information of the contextual situation of the item experience, for more than 27,000 POIs. This resulted in poor recommendations. Hence, we developed a novel rating elicitation method that evaluates the available items, their ratings and information about the users (their personality) in order to selectively propose to each user a personalised set of items to rate: these are estimated to be the most useful ones for the overall system performance improvement. This item selection strategy is an

[1] https://foursquare.com
[2] http://www.yelp.com
[3] http://www.pandora.com

example of Active Learning applied to recommender systems [32]. In active learning, the learner, which is here the recommender system, does not simply use the existing data (ratings) to learn the task (recommendation) but it actively searches for data (by making rating requests) that more effectively and efficiently help the learner to perform the task.

Due to the extreme cold-start situation, standard active learning strategies strive to correctly estimate the usefulness of the items as well as to acquire useful ratings from the users. Therefore, the proposed strategy collects and exploits some additional users information, i.e., their personality, in order to more effectively acquire not only ratings for the items, but also the contextual conditions that the user experienced while visiting the items. We illustrate and discuss the whole human-computer interaction that mainly includes: personality acquisition, target context of the recommendation setting, recommendation generation and browsing and rating elicitation with active learning. We show that the user personality is acquired with a simple and even engaging questionnaire that any user can fill out once in the registration process. The assessed personality can also be illustrated to the user hence making the interaction with the recommender even more rewarding for the user. We note that recommendations can even be computed relying only on the knowledge of the user personality. Hence, recommendations can be visualised right after the personality questionnaire is filled. But, using also the user ratings the recommendations are more precise. For that reason the user is prompted to enter more ratings while browsing the recommendations.

In conclusion, we list the main contributions of this article:

– We summarise and compare the results of several online user studies that we have conducted during the test of STS (see Section 6).
– We extend the analysis of the system's performance, which we have already conducted [4], by evaluating the system *Precision* (see Section 6).
– We illustrate the novel interface design with its improved representation of the user personality, the interface for context setting, the user profile page, and the messages inviting the user to enter more information (personality or ratings) (see [6, 12] for the description of previous interface design). For this novel design we have considered the users' feedback and analysed their interactions with the system (see Section 3).

The rest of this article is organised as follows. Section 2 discusses related work on the cold-start problem in CARSs. Section 3 illustrates the STS function and a typical interaction with the system. Section 4 and 5 describe in detail the implemented recommendation algorithm and the considered active learning strategies. Section 6 presents the designs of the conducted user studies and their results. Finally, conclusions are drawn and future work directions are described in Section 7.

2. Related work

While there has been a lot of research on the cold-start problem in traditional RSs [25, 34], only few studies have examined it specifically in CARSs. One direct solution is to acquire more information about the users, items or contextual situations. However, acquiring such information usually requires some additional effort of the users and/or service providers. This effort can be mitigated by using in the initial preferences acquisition phase an active learning strategy that, using some heuristics, identifies the items that the user is requested to provide feedback in terms of "likes", ratings, comments, etc [13]. Several heuristics have been used in the past, e.g., ask to rate the more popular items, or the items with more diverse ratings. The item selection criteria is only a heuristics since it is impossible to know, before having acquired a rating, first, if the rating will be really acquired (the user can rate the item or skip it) and second, what will be the rating value, and hence if it will tell something new about the user preferences that the system does not already know. Hence, an active learning strategy can only try to identify the items whose ratings will best reveal the user's preferences and thus will lead to better subsequent recommendations. [13] describes the fast growing literature on active learning for recommender systems and various item selection heuristics that have been proposed.

Another group of techniques, which are used for obtaining additional information and for starting a system even when few preference data (ratings) are available are cross-domain solutions. In these approaches, the system exploits additional preference data collected in auxiliary domains, where for instance a new user of the target system has revealed her preferences. So, for instance, one may rely on user preferences for movies for determining what types of books to recommend, even if no knowledge of the target user book-related preferences is available. The challenge is how to exploit this preference data related to items belonging to a domain that is different from the target one in order to improve the recommendation process in the target domain [15].

Another alternative solution to the cold start problem relies in exploiting metadata describing the users and the items (e.g., demographics and item descriptions), and to utilise them in a hybrid CARS [35]. Finally, it is also possible to rely on survey-based approaches and asking users to rate items in selected imaginary contexts, e.g., "Imagine that you can only use public transport. How likely is that you will visit Castel Flavon – Haselburg?", as it was done by Baltrunas et al. [2]. It has been shown that even if the contextual situation is only imagined the recommendations computed by using rating data in imagined contexts are more effective than those computed without relying on such information.

Instead of acquiring new preference data or metadata, one can also try to better process the existing preference data. In fact, an example of such approach is Differential Context Weighting (DCW) [36]. It is a pre-filtering approach [1] that instead of building a rating prediction model for each contextual condition (exact pre-filtering), combines ratings acquired in different contextual conditions by weighting them according to their similarities to the current target context. Another similar algorithm is Semantic Pre-Filtering (SPF) proposed by [10]. It is similar to the previous approach but uses a special notion of similarity based on distributional semantic.

In our research, among all these options, we explored the application of active learning in order to effectively acquire ratings from new users under different contextual situations. Our developed active learning strategy exploits user's personality information – using the Five Factor Model (FFM) [11] – in order to identify POIs whose ratings are useful and it is probable that the user experienced them. Personality is a predictable and stable factor that influences human behaviour. It has been shown that there exist direct relations between personality and tastes / interests [30], i.e., people with similar personality tend to exhibit similar interests and tastes. The implications of these findings were considered in research on personality-based CF systems [19]. However, to the best of our knowledge, no previous work has attempted to incorporate the user's personality in active learning.

We must also note that during the design of STS we have considered the lessons learned in the development of another mobile CARS application called ReRex [2]. In particular, we reused, and slightly adjusted, the

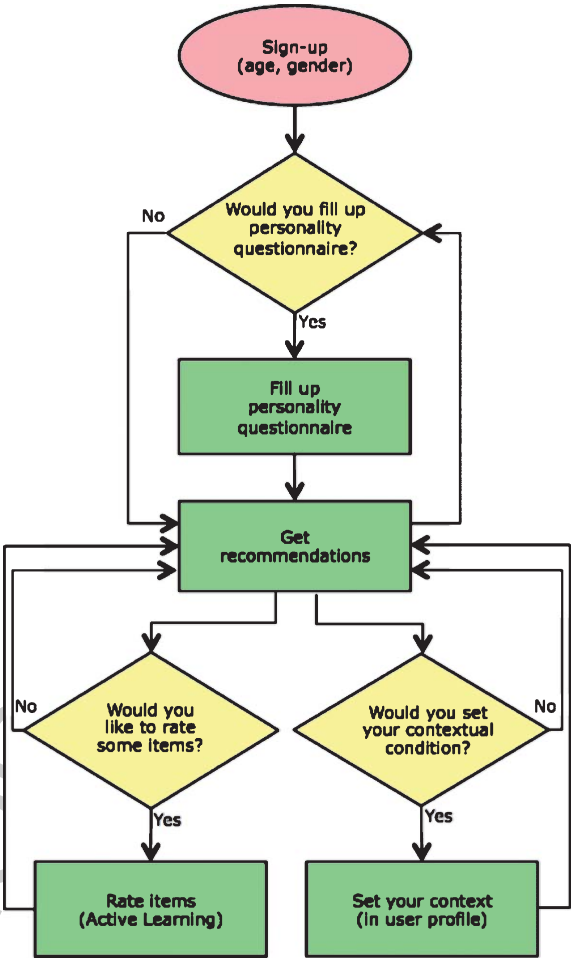

Fig. 1. Flowchart of HCI in STS.

set of contextual factors and contextual conditions that were found to be worth considering when generating POI recommendations and we reused and modified their context-aware matrix factorisation recommendation algorithm. We have extended that algorithm by incorporating additional user attributes (i.e., gender, birth date and personality trait information). This extension, as it is shown in this article, allows to produce personalised recommendations based on the aforementioned user attributes, even if the target user has not rated any items yet.

3. User interaction with the system

This section describes a typical system-user interaction with South Tyrol Suggests (STS), our

Android-based recommender system that provides users with context-aware recommendations for accommodations, attractions, events, public services and restaurants for the South Tyrol region of Italy. An overview of the user experience flowchart is depicted in Figure 1, where the oval denotes the start state, rectangles indicate actions the user takes (e.g., filling up the personality questionnaire), and diamonds are decisions the user makes (e.g., deciding whether or not to rate some items).

3.1. Personality questionnaire

After the user has registered to STS by entering her username, password, birthdate and gender, she is asked to complete the Five-Item Personality Inventory (FIPI) [17], so that the system can measure her Big Five personality traits (openness to experience, conscientiousness, agreeableness, extraversion and neuroticism). This is the first stage of the active learning procedure implemented in STS. It is used for building a predictive model of the user ratings, even without any information coming from other user ratings, and for identifying the items that the user should rate. The second stage refers to rating acquisition and it is illustrated in Section 3.3.

Figure 2 (left) shows a screenshot of our application where one of the questionnaire statements is illustrated. The full FIPI consists of the following five questions which require a 7-point Likert response ranging from “strongly disagree” to “strongly agree”:

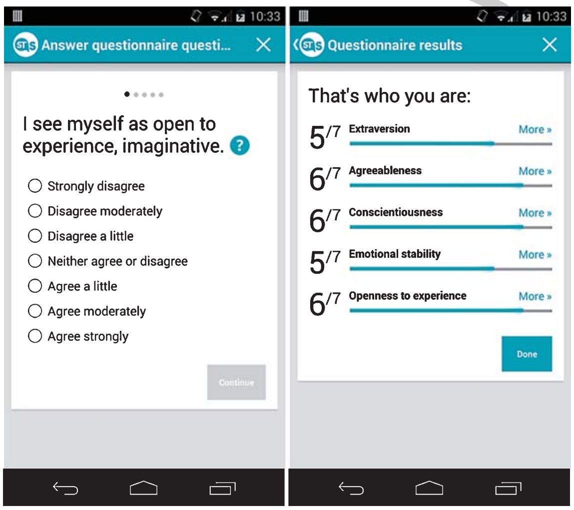

Fig. 2. Personality questionnaire.

1. I see myself as open to experience, imaginative;
2. I see myself as dependable, organized;
3. I see myself as extraverted, enthusiastic;
4. I see myself as agreeable, kind;
5. I see myself as emotionally stable, calm.

Since these questions may be difficult to understand, the application provides users with on-screen help including term definitions that can be accessed by clicking the question mark symbol next to each question, as can be seen in Figure 2 (left).

3.2. Recommendations

Using the assessed personality (as illustrated in Figure 2, right), the user’s age and gender (if available), and the value of 14 contextual factors, which are described in Section 3.4, the system identifies and shows a list of 20 highly relevant places of interest (POIs) (see Figure 3, left). We note that even though at this stage of the interaction no ratings of the user are known by the system, it can nevertheless generate personalised recommendations. In fact, the system runs a learning procedure that computes the parameters of a rating prediction model that determines how the user preferences depend on her personality (see Section 4 for details on the recommendation algorithm).

In the event the user is interested in one of these POIs, she can click on it and access the POI details window, as illustrated in Figure 3 (right). This window shows various information about the selected POI, such as a photo, its name, a description, user reviews, its category as well as an explanation of the recommendation based on the system estimated most influential contextual condition. Further supported features are, among others, the offered possibility to write a review for the POI, to request a route suggestion for reaching the POI from the current location, to tag the POI and to bookmark the POI, which makes it easy to get back to it later.

Another particularly interesting feature of the POI suggestions screen is that it provides users with two types of pop-up windows with information about how to obtain better recommendations. The first one, as can be seen in Figure 3 (bottom part of the left screen), requests the user to provide (more) ratings; clicking OK, forwards the user to a screen where she is requested to rate some specific items that are identified by the active learning component (see Section 3.3). The second type of pop-up window requests the user to specify her current context, i.e., the contextual factors that cannot

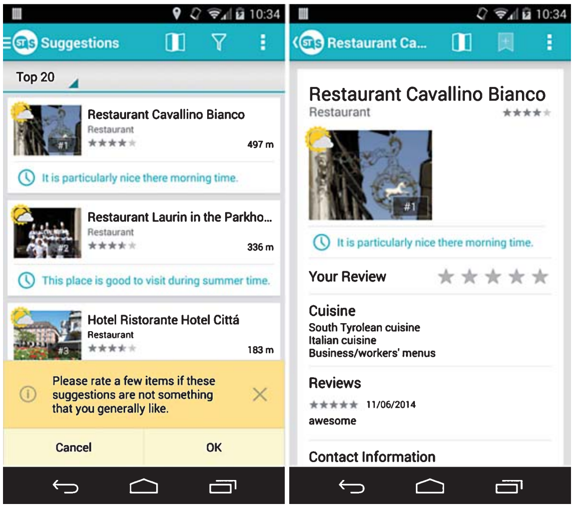

Fig. 3. Recommendations.

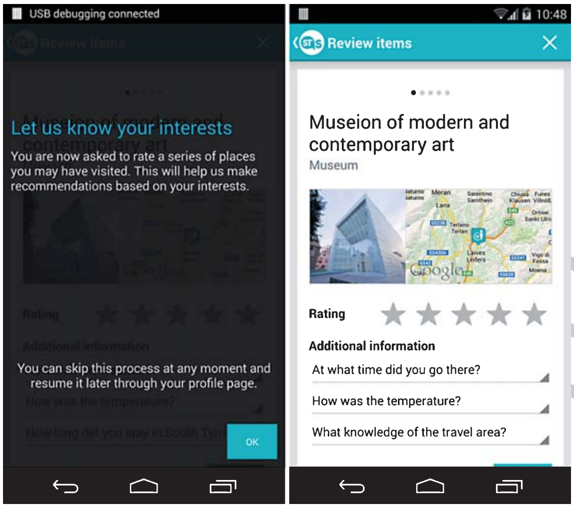

Fig. 4. Active learning.

be acquired automatically, such as budget, companion, transport. This is managed by an appropriate interface, as illustrated in more detail in Section 3.4.

3.3. Items rating

In order to start the rating acquisition interaction, the user is presented with a semi-transparent screen with a short explanatory text (see Figure 4, left). After dismissing this screen, the implemented personality-based binary prediction active learning module identifies 5 POIs that the system expects the user knows and can rate, and also whose ratings are useful for improving the quality of the subsequent recommendations (see Section 5 for the details on the active learning module). Figure 4 (right) shows a screenshot where the user is asked to rate a POI, if she has experienced it, and to specify the contextual situation of that experience, if she remembers and is eager to provide it. For instance, here, the user is asked to specify the time of the day of her visit, the temperature as well as her knowledge of the travel area. For each of the displayed POIs, the user can specify the value of up to three contextual factors, describing the contextual situation of the user when she experienced the POI. The three displayed factors are selected from the full set of 14 factors managed by system (see next section) by, first scoring all the factors according to their relevance in the rating prediction model [2], and then sampling at random three contextual factors from the full list with a probability that is proportional to the relevance score. We decided to request the user to specify only three contextual factors because a larger number would not fit the screen of a mobile phone. However, the implemented random sampling of the contextual factors allows to collect data describing the impact of all the various contextual factors on the ratings.

3.4. Context settings

The context settings are accessible from the user profile page, as illustrated in Figure 5 (left). They allow the user to fine-tune her current contextual situation by

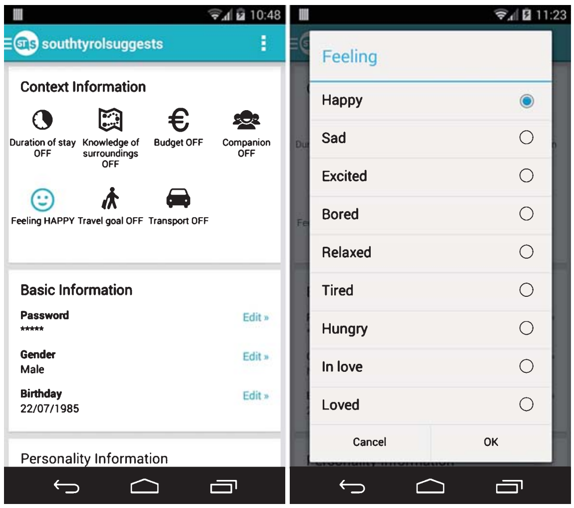

Fig. 5. User profile.

Table 1
Contextual factors used in the STS application

Contextual factors	Associated contextual conditions
Weather	Clear sky, sunny, cloudy, rainy, thunderstorm, snowing
Season	Spring, summer, autumn, winter
Budget	Budget traveler, high spender, none of them
Daytime	Morning, noon, afternoon, evening, night
Companion	Alone, with friends/colleagues, with family, with girlfriend/boyfriend, with children
Feeling	Happy, sad, excited, bored, relaxed, tired, hungry, in love, loved, free
Weekday	Working day, weekend
Travel goal	Visiting friends, business, religion, health care, social event, education, scenic/landscape, hedonistic/fun, activity/sport
Transport	No transportation means, a bicycle, a motorcycle, a car, public transport
Knowledge of the travel area	New to area, returning visitor, citizen of the area
Crowdedness	Crowded, some people, almost empty
Duration of stay	Some hours, one day, more than one day
Temperature	Burning, hot, warm, cool, cold, freezing
Distance	Far away (over 3 km), nearby (within 3 km)

enabling and setting the values of those factors that can not be automatically acquired, such as the duration of the current stay, the user knowledge of the travel area, the current budget, the actual companion and feelings. The full set of contextual factors and contextual conditions has been derived from Baltrunas et al. [2] and can be found in Table 1. The contextual factors Daytime, Weekday, Distance (to POI), Weather, Season, Temperature and Crowdedness are automatically obtained and are not entered by the user. The remaining contextual factors, if the user has enabled them, must be entered manually by the user, as displayed in Figure 5 (right).

4. Recommendation algorithm

Recommendation algorithms are machine learning techniques that given a data set of ratings for items predict the missing ratings of users for the items that they have not rated. Then, given these predictions the items with the highest predicted ratings are recommended. The underlying assumption is that the items that the user has not yet rated may be novel for the user and if the system predicts that the user would rate them high then they may be also relevant suggestions. Hence, the core step of a recommendation algorithm is rating prediction.

We will denote with $u \in U$ a user, and with $i \in I$ an item. r_{ui} is the rating that the user u gave to the item i and $\hat{r}_{uj}$ is the rating predicted by a model for an item j whose rating r_{uj} is unknown. e_{ui} is the prediction error, i.e., the difference between $\hat{r}_{ui}$ and r_{ui} for user u and item i. This assumes that r_{ui} is known but not used in building the prediction model. Learning algorithms use the measured errors in order to update model parameters and ultimately reduce the error.

Matrix factorisation models are the most widely used technique for building CF prediction models [22]. In a matrix factorisation model each item i and user u is associated with an f-dimensional real vector q_i and p_u. The elements of $q_i = (q_{i1}, \dots, q_{if})$ measure the extent to which the item i possesses those factors, positive or negative. The elements of $p_u = (p_{u1}, \dots, p_{uf})$ measure the extent of interest u has in items that are high on the corresponding factors, positive or negative. These factors are said to be "latent" because they are not observable properties of the items or the users but are computed (learned) by the system with the only goal of improving the system's rating prediction accuracy, i.e., trying to make rating predictions as close as possible to observable ratings. The interaction between user u and item i is captured by the dot product $q_i \cdot p_u$. This dot product given in Equation 1 produces a estimation of the users' overall interest in the items based on the latent features used in the model [22].

$$\hat{r}_{ui} = q_i \cdot p_u = \sum_{j=1}^{f} q_{ij} p_{uj} \tag{1}$$

More complex and reliable models introduce other parameters in addition to the user and item vectors. We will illustrate below our precise model. The model parameters, i.e., the vector representations of the users and items (and other parameters if present), are learned by minimising the error of the model predictions on a training set of ratings [22]. In our experiments, as it is usual in these approaches, we use one of the state of the art model learning techniques, i.e., minimising regularised squared error with Stochastic Gradient Descent. Its idea is looping through ratings in the training set, predicting rating $\hat{r}_{ui}$, computing the associated prediction error e_{ui} and changing the model parameters moving in the opposite direction of the gradient of the regularised error.

In our system we have further extended the matrix factorisation approach for rating prediction with contextual information that was proposed by Baltrunas et al. [2]. Our prediction model, which is illustrated below, incorporates baseline parameters for each contextual condition and item combination, besides the standard parameters used in context-free matrix factorisation, i.e., global average, item bias, user bias and user-item interaction. A baseline parameter is a model parameter that is computed by the model learning algorithm and describes the influence on the rating of a specific element of the recommendation scenario. For instance, the user bias is a parameter that describes how a particular user rates, on average, the items. Larger user bias values indicate that the user is typically assigning larger ratings to items.

Moreover, since the original context-aware matrix factorisation model fails to provide personalised recommendations for users with no or few ratings (i.e., new user problem), we also enhance the representation of a user by introducing into the model parameters that represent the known user attributes: age group, gender and the scores for the Big Five personality traits. Our approach follows an analogous solution described in [23]. This allows to model the user preferences, even in cases where implicit and explicit feedback are absent.

The resulting model computes a rating prediction for user u and item i in the contextual situation described by the contextual conditions c_1, ..., c_k using the following rule:

$$\hat{r}_{uic_1,\ldots,c_k} = \bar{i} + b_u + \sum_{j=1}^{k} b_{ic_j} + \; q_i \cdot (p_u + \sum_{a \in A(u)} y_a), \tag{2}$$

where q_i, p_u and y_a are f-dimensional real-valued vectors representing the item i, the user u and the user attribute a, respectively. $\bar{i}$ is the average rating for item i, b_u is the baseline parameter for user u and b_{ic_j} is the baseline parameter that models how the contextual condition c_j influences the rating of item i.

Model parameters are learned offline, once every five minutes, by minimising the associated regularised squared error function through stochastic gradient descent [22]. The learning procedure is fast and only few seconds are needed to re-train the model when new information is acquired (new ratings or new users).

5. Active learning strategies

We describe in this section the active learning strategies that we have used in STS. We implemented and compared four strategies.

The first one is *log(popularity) * entropy*. It is a state-of-the-art solution and it is used as baseline for the comparison. We have used this particular baseline since previous works have compared it with other competing approaches and reported its excellent performance [14, 26, 28, 29]. log(popularity) * entropy scores each item i by multiplying the logarithm of the *popularity* of i (i.e., the number of ratings for i in the training set) with the entropy of the ratings for i. Then, the top items according to the computed score (4 in our experiments) are proposed to be rated by the user. This strategy tries to combine the effect of the popularity with a score that favours items with more diverse ratings (larger entropy), which may provide more useful (discriminative) information about the user's preferences [8, 28]. Clearly, popular items are more likely to be known by the user, and hence it is more likely that a request for such a rating will be fulfilled by the user and will increase the size of the rating database. But many popular items in our dataset had no or only one rating, and rating-based popularity scores cannot distinguish such popular items from less popular ones with similar number of ratings. Therefore, this strategy may fail to select the true popular items and may suggest items that are unknown to the user and thus not rateable.

To cope with that problem, we have designed our personality-based strategy that tries to better identify the items that the user has experienced, by exploiting the personality information of the users. *Personality-Based Binary Prediction* first transforms the rating matrix to a matrix with the same number of rows and columns, by mapping null entries to 0, and not null entries to 1. Hence, the new matrix models only whether a user rated an item or not, regardless of its value. Then, the new matrix is used to train an extended version of the matrix factorisation algorithm. The model that we have used is similar to that used for rating prediction but it does not use contextual information. It profiles users in terms of the binary ratings (rated vs. not rated item), and using known user attributes, in our case, gender, age group and the scores for the Big Five personality traits on a scale from 1 to 5. Given a user u, an item i

and the set of user attributes $A(u)$, it predicts a user-item score using the following rule:

$$\hat{s}_{ui} = \bar{i} + b_u + q_i \cdot (p_u + \sum_{a \in A(u)} y_a), \qquad (3)$$

where p_u, q_i and y_a are the latent factor vectors associated with the user u, the item i and the user attribute a, respectively. The model parameters are then learned, as it is common in matrix factorisation, by minimising the associated regularised squared error function through stochastic gradient descent. The learned model predicts and assigns a rateable score to each candidate item i (for each user u), with higher scores indicating a higher probability that the target user has consumed the item i, and hence may be able to rate it.

In the live user studies that are described below log(popularity) * entropy and Personality-Based Binary Prediction have been also used in a "compound" form. This compound strategy was implemented by letting each strategy to suggest a short list of items to rate (4 in our experiment) and then merging the two lists. In general, exploiting a combination of strategies with different characteristics is beneficial: the merits of both strategies can be exploited, and a combination of strategies adds some diversity to the system's rating requests. Moreover, even though the two strategies were combined, we were still able to compare their individual performances in an offline analysis where we built separated training sets with the ratings acquired by each individual strategy, which is an information that was logged during the experiments.

Finally, in order to have an additional baseline we have also considered a fourth strategy, that simply requests to rate items selected at *Random*.

6. Live user studies and results

In order to assess the full performance of our mobile recommender in a cold-start situation, and compare the proposed techniques with state-of-the-art solutions, we have formulated a number of hypotheses, and designed and performed several live user studies. Here we list the hypotheses, while the conducted user studies are presented in the following subsections (see Table 2 for a summary of the experiments and results).

We have formulated the following hypotheses that we conjecture hold in the cold-start situation of a CARS when the proposed system is compared to a system that is using the same interaction design but selects and asks the user to rate items identified by a baseline active learning approach, namely log(popularity) * entropy, which is fully described in the previous section.

- H1 The proposed personality-based active learning (AL) strategy collects ratings that let the system achieve higher recommendation accuracy.
- H2 The proposed AL strategy leads to a higher number of ratings acquired from users.
- H3 The proposed AL strategy can acquire a larger number of contextual conditions, describing the experience of the user at the visited POIs.
- H4 The proposed recommendation model, trained on ratings elicited by the proposed AL strategy, recommends POIs that better suit the user's contextual situation.

6.1. Experiment 1: Prediction accuracy and number of acquired ratings

In a first experiment [12] our goal was to study the influence of the rating elicitation strategies on the evolution of the RS's performance and of the ratings data set. 108 participants were randomly assigned to two groups: one half was assigned to the compound AL strategy (n = 54) and the other half to the random AL strategy (n = 54). These strategies are described in detail in Section 5. Users of both groups were first asked to complete the personality questionnaire and then to rate some train items selected by their assigned AL strategy and some test items randomly selected among the remaining items. Then, off-line, after having trained the prediction model on all the training ratings acquired from the users with a specific AL strategy during the study, the Mean Absolute Error (MAE), i.e., absolute deviation of the real ratings (acquired in the live user study) and predicted ratings for the random test items was measured. We note that the test items were selected randomly since we were interested in measuring the unbiased accuracy of the rating prediction model, i.e., not only on items estimated to be "relevant" (and that could be recommended), but also on items that were estimated to be "irrelevant" (and should not be recommended). In another experiment, which is described in the next section, the test items were only those recommended by the system, which is a more popular approach in recommender systems.

Mean Absolute Error (MAE): We have first measured the MAE of the system trained only on the ratings that were available before conducting this experiment. This initial MAE was 1.06 and it is an indication of the

Table 2
Summary of the evaluations and results

Experiment	Metric	# of Participants	Results
1) Recommendation accuracy [12]	MAE	108	personality based strategy achieved the lowest MAE and elicited biggest # of ratings
2) Context-awareness [4]	feedback on items — # of acquired contexts — Precision	51	personality based strategy achieved the highest scores for context-awareness and precision and elicited biggest # of contexts
3) Recommendation quality [6]	perceived rec. quality — choice satisfaction	54	weather-aware system achieved the highest perceived recommendation quality and choice satisfaction
4) System usability [5]	SUS score	30	the overall system achieved a high usability score (well-above the SUS benchmark)

accuracy of the prediction model before the application of active learning. Afterward, i.e., by adding to the training set and retraining the prediction model with the ratings acquired by the personality-based strategy the MAE estimated on the test items dropped to 0.86. Whereas the MAE was 0.90 after adding the ratings collected by the log(popularity) * entropy strategy, and 0.97 by adding the ratings acquired by the random strategy. Hence, the MAE reduction, due to the AL, is 18.8% for personality-based binary prediction, 15.0% for log(popularity) * entropy, and 8.4% for random. Evidently, under the application of our proposed strategy, the system's MAE was reduced the most [12]. This supports the H1 hypothesis.

Number of Acquired Ratings: We have also measured the number of ratings that were acquired by the considered strategies. As we discussed before, certain strategies can acquire more ratings by better estimating what items are likely to have been experienced or known by the user. Hence, the number of acquired ratings can indicate the goodness of a strategy in the estimation of the rateability of the items. We note that before performing active learning, there were 848 ratings available in our database. The personality-based strategy acquired 125 ratings, while 112 were acquired by log(popularity) * entropy strategy and 73 by random strategy.

Overall the average number of ratings acquired per user was 1.9 (out of 4 items that were requested to rate in the training set). The personality-based binary prediction strategy elicited on average 2.31 ratings from each user, whereas the log(popularity) * entropy strategy elicited 2.07 ratings, and the random strategy elicited 1.35 ratings. Hence, the personality-based binary prediction strategy can elicit significantly more ratings than the two competitor strategies. This result supports the H2 hypothesis.

6.2. Experiment 2: Quality of context-dependent recommendations

In a second experiment [4] the goal was to evaluate the quality of the individual recommendations produced by our context-aware recommender. In contrast to the previous experiment, where the test items were selected randomly, in this experiment, the test items were those recommended. In total 51 participants have taken part in the experiment. The users were randomly assigned either to the log(popularity) * entropy strategy group (n = 19) or the personality-based binary prediction item selection strategy group (n = 27). Some users from log(popularity) * entropy strategy group were then excluded because they did not complete the evaluation. Each user was prompted to rate 8 items, selected by the assigned strategy, and then the system generated 4 personalised recommendations. The system also asked the user to evaluate each of the recommended POIs, by means of dialogue window that popped up after doing a long press on them (see figure 6). The users were instructed to do that. The dialogue window contained the following two specific statements to be answered on a five-star rating scale (1 star being the lowest score and 5 stars being the highest score):

– *Q1: Does this recommendation fit my preference?*
– *Q2: Is this recommendation well-chosen for the situation?*

These statements were obtained from [21], which provides a standard questionnaire for perceived recommendation quality and choice satisfaction. We chose these statements since they assess two important features of a recommender system, i.e., how the recommendations fit the preference of the user (general preference) and how relevant and well-chosen they are

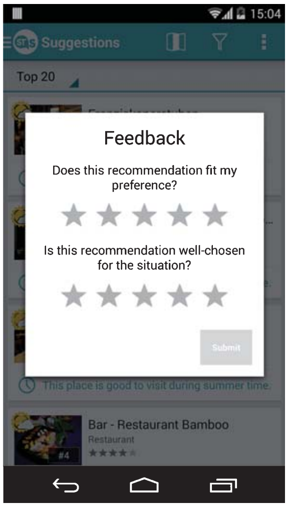

Fig. 6. User feedbacks for two questions.

for the specific user situation (specific needs related to the context).

Recommendation Context-Awareness: The users assigned to personality-based binary prediction Active Learning replied with an average score of 3.56 to Q1 (preference fitting) and 3.31 to Q2 (context awareness). In the group of users assigned to the log(popularity) * entropy strategy, these average scores are 3.58 (for Q1) and 2.95 (for Q2), respectively. We observe that both strategies obtained almost the same average reply to Q1 (no significant difference, $p = 0.43$ for a t-test), while the personality-based strategy obtained a significantly higher average score for Q2 ($p = 0.049$). Therefore, while both strategies acquired ratings that resulted in recommendations that "fit the preferences" of the users, the proposed personality-based binary prediction strategy outperforms log(popularity) * entropy, by acquiring ratings that result in recommendations that are evaluated to be more "well-chosen" for the contextual conditions of the users [4]. This result supports the hypothesis H4.

Number of Acquired Contextual Conditions: We have also counted the number of acquired contextual conditions that each user entered during the active learning process. As we explained before, when the POIs are presented to the users to rate, the users could also enter the contextual conditions they experienced when visiting the POIs, if they were eager to provide this information. We therefore compared the active learning strategies in terms of how many contextual conditions were entered by the users, in order to describe their experience, during the rating elicitation process. We observed that the

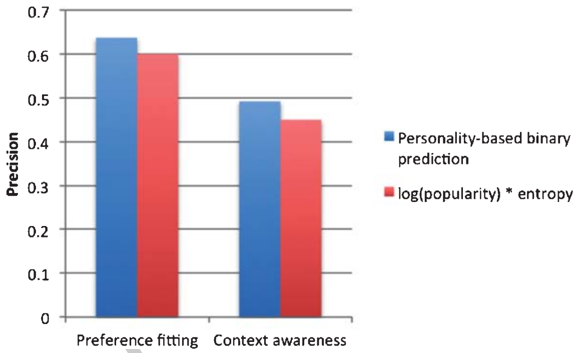

Fig. 7. Precision @ 4.

users that were assigned to the personality-based active learning strategy, by average, entered 1.52 contextual conditions (out of 3) vs. 1.01 entered by the users assigned to log(popularity) * entropy strategy variant ($p < 0.001$). Hence, our proposed personality-based active learning strategy acquires a significantly larger number of contextual conditions in comparison to the state-of-the-art log(popularity) * entropy. We believe that this result is due to the fact that the personality-based strategy selects POIs that are more familiar to the users and hence users may better remember the experience of their visit (and the contextual conditions) [4]. This result supports the research hypothesis H3.

We have also measured the *Precision* of the system. A recommendation evaluated by the user, replying to the questions Q1 and Q2, equal or above 4 is considered relevant, while below 4 irrelevant. We have then computed the precision@4 metric for each user and then averaged it over all users. Figure 7 shows that the recommender trained with the ratings acquired by the personality-based binary prediction strategy has precision 0.63 for Q1 (preference fitting) and 0.49 for Q2 (context-awareness). When the system is trained with the ratings acquired by the log(popularity) * entropy strategy, these values are smaller, 0.60 and 0.45, respectively. We should note that, while these are interesting results, these performance differences have no statistical significance. We believe that this is due to the small sample of test participants.

6.3. Experiment 3: Weather awareness

In the third experiment [6] the goal was to assess the importance of a particular contextual factor, the weather conditions, which has been shown to be among the most influential factor in tourism decision making [24]. In fact, the weather conditions affect considerably the user's perceived suitability of a POI. In order to assess whether our management of the weather conditions was effective, we designed an experiment with a tailor made user task and we used the questionnaire proposed by [21] that is designed for assessing the perceived recommendation quality and choice satisfaction. 54 participants were randomly divided in two equal groups assigned to either STS (n = 27) or to a similar variant called STS-S (n = 27). While both variants have similar interfaces, they differ in the way the weather factor is used in the recommender system. More precisely, STS has a user interface where the weather forecast is shown (missing in STS-S) and it exploits the weather condition at the item location for better predicting items' ratings (missing in STS-S).

The users were invited to imagine that they had an afternoon off, to look for attractions or events in South Tyrol, to consider which contextual conditions were relevant for them and to specify them in the system settings. Afterwards, they were invited to browse the system suggestions (recommendations), to select the one that they believed fitted their needs and wants and bookmark it. Finally, the users filled up a survey [21], which contains the following statements:

Perceived recommendation quality:

- Q1: I liked the items suggested by the system.
- Q2: The suggested items fitted my preference.
- Q3: The suggested items were well-chosen.
- Q4: The suggested items were relevant.
- Q5: The system suggested too many bad items.
- Q6: I didn't like any of the suggested items.
- Q7: The items I selected were "the best among the worst".

Choice satisfaction:

- Q8: I like the item I've chosen.
- Q9: I was excited about my chosen item.
- Q10: I enjoyed watching my chosen item.
- Q11: The items I watched were a waste of my time.
- Q12: The chosen item fit my preference.

After this initial interaction, we wanted to assess whether an explicit reference to the weather conditions at the selected POI may push the user to change her selection. Hence, we offered the user the opportunity to double check the weather conditions at the selected POI by accessing on a computer the Mondometeo[4] website. This was offered to the users in both groups. Afterwards, we asked the users whether they wanted to change their preferred POI and bookmark another one, i.e., if they believed that, because of the weather conditions at the selected POI, their previous choice was not anymore appropriate. If a user changed the preferred POI we asked her to evaluate the new POI that was selected, by answering the following two questions:

- Q14: I like the new item I've chosen.
- Q15: Please say again whether the suggested items were well chosen.

We analysed all the survey data in order to discover which system overall scored better with respect to users' perceived recommendation quality and users' choice satisfaction. We computed a score for each system by averaging the total score obtained by summing up the answers given by each user. For STS-S we obtained an average (per user) score of 22.8 for the users' perceived recommendation quality and 19.5 for the users' choice satisfaction. For STS these numbers were 24.7 and 19.8. Hence, in both cases STS achieved a higher performance. Moreover, we found that almost 60% of the users (16 users out of 27) who have used STS-S have changed their POI selection after the assessment of the weather conditions, while only 30% of the STS users (8 users out of 27) changed their selection. The p-value of the chi-square test used for comparing these two proportions is equal to 0.028. Hence, significantly more users that selected their preferred POI using STS-S, i.e., with a system that does not take into account the weather conditions in its recommendations, revised their decision after having realised that the weather at the POI made their selection not really optimal. These results all indicate a high perceived recommendation quality and user's choice satisfaction with our STS app [6].

6.4. *Experiment 4: Usability test*

Finally, in a forth experiment [5] we assessed the overall system usability. We have chosen SUS (System Usability Scale) [7] that has become a standard for user experience and usability analysis. Using SUS a system is scored with a positive number smaller or equal to 100. The commonly used benchmark for the comparison of the usability of the system is 68. This was computed in a study of 500 software systems conducted by [33]. In fact, this is a strong baseline for our mobile RS, since this benchmark was established on standard PC-based applications, and not on mobile systems. Mobile applications pose additional usability issues. For a mobile application it is harder to achieve that benchmark score as it requires to deal with the

[4]http://mondometeo.org

significant variation among mobile devices such as differences in screen size, screen resolution, CPU performance characteristics, input mechanisms (e.g., soft keyboards, hard keyboards, touch), memory and storage space and installed fonts.

30 participants completed the test and analysing the results we have observed that STS obtained an average SUS score of 77.92, that is well above the benchmark of 68. It has been shown that this SUS score falls between "good" and "excellent" (in terms of the adjectives that the users may use to evaluate the system) [3].

In conclusion, the results obtained from different user studies, with a variety of test methodologies, and a broad range of evaluation metrics, have confirmed that the proposed active learning techniques can be effectively used to overcome the cold-start problem in mobile context-aware recommender systems, especially in the tourism domain.

7. Discussion and future work

In this paper, we have presented practical solutions to overcome the cold-start problems encountered in CARSs, i.e., the shortage of ratings for items in alternative contextual situations. We have developed a novel active learning strategy as well as a context-aware recommendation algorithm that both make use of an extended matrix factorisation model for incorporating the user's personality in terms of of the Five Factor Model (FFM) (i.e., openness, conscientiousness, extroversion, agreeableness and neuroticism), and for providing new users with personalised rating requests and recommendations, respectively. Our developed solutions were all integrated into our Android-based points of interest recommender system, called South Tyrol Suggests (STS), which in the early stages of its deployment was suffering from an extreme cold-start situation where only a few hundred contextually-tagged ratings were available for the approximately 27,000 POIs stored in the system database, thus resulting in poor recommendations.

While CARSs have been mostly evaluated offline, in our analysis we have evaluated the proposed solutions in several live user studies aimed at testing a collection of research hypotheses (see Section 6). In summary, the results obtained from our user studies confirm that the proposed preference elicitation procedure, based on the exploitation of the user personality in an active learning component, and the interaction design developed for supporting the preference elicitation process, are helpful in producing effective recommendations even in the cold start situation.

We should note that our research is focussed on the travel and tourism domain, which is a particularly complex domain compared with other domains, such as music and movies. In fact, active learning methodologies for tourism applications must be different from those adopted in other domains such as music. One reason is related to user experience: in the music domain the system can ask the user to rate a song that she has not listened before, just by offering the user an excerpt. This is not feasible in the tourism domain; the user can only rate a POI if she has visited it or has a really good knowledge of it.

For future work, we plan to continue experimenting with novel approaches to overcome the cold-start problem associated with CARSs, such as our STS application.

Firstly, we intend to perform an extensive analysis of the applicability of hybrid techniques where several basic recommendation techniques are combined, each one having its own strengths and weaknesses. The goal is to produce meaningful recommendations also in cold-start situations when non-hybrid and basic recommenders alone would fail. In particular, we focus on the following research questions: 1) which basic CARS algorithms should be combined into a hybrid one?, and 2) how should the basic CARS algorithms be hybridised to handle the various cold-start situations?

Additionally, we would like to examine whether the knowledge of the user emotions and current activity, which can be automatically derived from wearable devices such as smartwatches and smartbands, can be used in the recommender system and impact on the system accuracy, in particular in cold-start scenarios. This involves developing a clear understanding of these wearable devices, their sensor data as well as derivable emotional states or activities.

Another important future work is to develop strategies to actively select the contextual factors that are most informative and relevant for the user's rating of a specific POI. The importance of this task lies in the fact that not all contextual factors are equally useful for the system – for certain POIs (or POI categories) some contextual factors are much more informative and important than others. For instance, the weather factor is more important to assess the relevance of outdoor POIs (e.g., lakes, mountain excursions, scenic walks) rather than indoor POIs (e.g., museums, churches, castles) since it is expected to have a bigger impact on the user ratings.

Finally, we plan to explore the role of proactivity in helping to solve the cold-start problem associated with CARSs. A proactive recommender system actively pushes recommendations or rating requests to the user when the current context seems appropriate. Despite the advantages of proactive systems, especially in the mobile scenario where users could be provided with recommendations or rating requests on the fly as they need them, the area of proactive recommender systems is still an unexplored field with many open problems. For instance, it is necessary to determine whether and, if yes, when the users are willing to accept proactive notifications. Moreover, it requires to identify how to interrupt the user and present her with the information.

Acknowledgements

The authors wish to acknowledge the anonymous reviewers for their detailed and helpful comments to the manuscript.

References

[1] G. Adomavicius, B. Mobasher, F. Ricci and A. Tuzhilin, Context-aware recommender systems, *AI Magazine* **32**(3) (2011), 67–80.

[2] L. Baltrunas, B. Ludwig, S. Peer and F. Ricci, Context relevance assessment and exploitation in mobile recommender systems, *Personal and Ubiquitous Computing* **16**(5) (2012), 507–526.

[3] A. Bangor, P. Kortum and J. Miller, Determining what individual sus scores mean: Adding an adjective rating scale, *Journal of Usability Studies* **4**(3) (2009).

[4] M. Braunhofer, M. Elahi, M. Ge and F. Ricci, Context dependent preference acquisition with personality-based active learning in mobile recommender systems. In *Learning and Collaboration Technologies. Technology-Rich Environments for Learning and Collaboration*, Springer International Publishing, 2014, pp. 105–116.

[5] M. Braunhofer, M. Elahi and F. Ricci, Usability assessment of a context-aware and personality-based mobile recommender system. In *E-Commerce and Web Technologies*, Springer, 2014, pp. 77–88.

[6] M. Braunhofer, M. Elahi, F. Ricci and T. Schievenin, Contextaware points of interest suggestion with dynamic weather data management. In *Information and Communication Technologies in Tourism 2014*, Springer, 2013, pp. 87–100.

[7] J. Brooke, Sus-a quick and dirty usability scale, *Usability Evaluation in Industry* **189** (1996), 194.

[8] G. Carenini, J. Smith and D. Poole, Towards more conversational and collaborative recommender systems. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, ACM, 2003, pp. 12–18.

[9] G. Chen and L. Chen, Recommendation based on contextual opinions. In *User Modeling, Adaptation, and Personalization*, Springer, 2014, pp. 61–73.

[10] V. Codina, F. Ricci and L. Ceccaroni, Local context modeling with semantic pre-filtering. In *Proceedings of the 7th ACM Conference on Recommender Systems*, ACM, 2013, pp. 363–366.

[11] P. Costa and R. McCrae, Toward a new generation of personality theories: Theoretical contexts for the five-factor model. *The five-factor model of personality: Theoretical perspectives*, 1996, pp. 51–87.

[12] M. Elahi, M. Braunhofer, F. Ricci and M. Tkalcic, Personalitybased active learning for collaborative filtering recommender systems. *In AI* IA 2013: Advances in Artificial Intelligence*, Springer, 2013, pp. 360–371.

[13] M. Elahi, F. Ricci and N. Rubens, Active learning in collaborative filtering recommender systems. In *E-Commerce and Web Technologies*, Springer, 2014, pp. 113–124.

[14] M. Elahi, F. Ricci and N. Rubens, Active learning strategies for rating elicitation in collaborative filtering: A systemwide perspective, *ACM Transactions on Intelligent Systems and Technology* **5**(1) (2014).

[15] M. Enrich, M. Braunhofer and F. Ricci, Cold-start management with cross-domain collaborative filtering and tags, In *E-Commerce and Web Technologies*, Springer, 2013, pp. 101–112.

[16] D. Gallego, W. Woerndl and G. Huecas, Evaluating the impact of proactivity in the user experience of a context-aware restaurant recommender for android smartphones, *Journal of Systems Architecture* **59**(9) (2013), 748–758.

[17] S.D. Gosling, P.J. Rentfrow and W.B. Swann Jr, A very brief measure of the big-five personality domains, *Journal of Research in Personality* **37**(6) (2003), 504–528.

[18] N. Hariri, B. Mobasher and R. Burke, Context-aware music recommendation based on latenttopic sequential patterns. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, ACM, 2012, pp. 131–138.

[19] R. Hu and P. Pu, Enhancing collaborative filtering systems with personality information. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, ACM, 2011, pp. 197–204.

[20] M. Kaminskas, F. Ricci and M. Schedl, Location-aware music recommendation using auto-tagging and hybrid matching. In *Proceedings of the 7th ACM Conference on Recommender Systems*, ACM, 2013, pp. 17–24.

[21] B.P. Knijnenburg, M.C. Willemsen, Z. Gantner, H. Soncu and C. Newell, Explaining the user experience of recommender systems, *User Modeling and User-Adapted Interaction* **22** (4-5) (2012), 441–504.

[22] Y. Koren and R. Bell, Advances in collaborative filtering. In F. Ricci, L. Rokach, B. Shapira and P.B. Kantor, editors, Recommender Systems Handbook, Springer Science and Business Media, 2011, pp. 145–186.

[23] Y. Koren, R. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, *Computer* **42**(8) (2009), 30–37.

[24] N. Kozak, M. Uysal and I. Birkan, An analysis of cities based on tourism supply and climatic conditions in turkey, *Tourism Geographies* **10**(1) (2008), 81–97.

[25] X.N. Lam, T. Vu, T.D. Le and A.D. Duong, Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, ACM, 2008, pp. 208–211.

[26] S.M. McNee, S.K. Lam, J.A. Konstan and J. Riedl, Interfaces for eliciting new user preferences in recommender systems. In *User Modeling 2003*, Springer, 2003, pp. 178–187.

[27] N. Natarajan, D. Shin and I.S. Dhillon, Which app will you use next?: Collaborative filtering with interactional context. In *Pro-*

ceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 201–208.

[28] A.M. Rashid, I. Albert, D. Cosley, S.K. Lam, S.M. Mcnee, J.A. Konstan and J. Riedl, Getting to know you: Learning new user preferences in recommender systems. *In Proceedings of the 2002 International Conference on Intelligent User Interfaces, IUI 2002*, ACM Press, 2002, pp. 127–134.

[29] A.M. Rashid, G. Karypis and J. Riedl, Learning preferences of new users in recommender systems: An information theoretic approach, *SIGKDD Explor Newsl* **10** (2008), 90–100.

[30] P.J. Rentfrow, S.D. Gosling, et al. The do re mi's of everyday life: The structure and personality correlates of music preferences, *Journal of Personality and Social Psychology* **84**(6) (2003), 1236–1256.

[31] F. Ricci, L. Rokach and B. Shapira, *Introduction to recommender systems handbook*, Springer, 2011.

[32] N. Rubens, D. Kaplan and M. Sugiyama, Active learning in recommender systems. In F. Ricci, L. Rokach, B. Shapira and P. Kantor, editors, *Recommender Systems Handbook*, Springer Verlag, 2011, pp. 735–767.

[33] J. Sauro, Measuring usability with the system usability scale (sus). http://www.measuringusability.com/sus.php. Accessed: 2013-01-15.

[34] A.I. Schein, A. Popescul, L.H. Ungar and D.M. Pennock, Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2002, pp. 253–260.

[35] W. Woerndl, M. Brocco and R. Eigner, Context-aware recommender systems in mobile scenarios, *International Journal of Information Technology and Web Engineering (IJITWE)* **4**(1) (2009) 67–85.

[36] Y. Zheng, R. Burke and B. Mobasher, Recommendation with differential context weighting. In *User Modeling, Adaptation, and Personalization*, Springer, 2013, pp. 152–164.