Usability Assessment of a Context-Aware and Personality-Based Mobile Recommender System

Matthias Braunhofer, Mehdi Elahi, and Francesco Ricci

Free University of Bozen, Bolzano,
Piazza Domenicani 3, Bolzano, Italy
{mbraunhofer,mehdi.elahi,fricci}@unibz.it
http://www.unibz.it

Abstract. In this paper we present STS (South Tyrol Suggests), a context-aware mobile recommender system for places of interest (POIs) that integrates some innovative components, including: a *personality questionnaire*, i.e., a brief and entertaining questionnaire used by the system to learn users' personality; an *active learning* module that acquires ratings-in-context for POIs that users are likely to have experienced; and a *matrix factorization* based recommendation module that leverages the personality information and several contextual factors in order to generate more relevant recommendations.

Adopting a system oriented perspective, we describe the assessment of the combination of the implemented components. We focus on usability aspects and report the end-user assessment of STS. It was obtained from a controlled live user study as well as from the log data produced by a larger sample of users that have freely downloaded and tried STS through Google Play Store. The result of the assessment showed that the overall usability of the system falls between "good" and "excellent", it helped us to identify potential problems and it provided valuable indications for future system improvement.

Keywords: Recommender systems, context awareness, mobile services, active learning, personality, usability assessment.

1 Introduction

Tourist's decision making is the outcome of a complex decision process that is affected by "internal" (to the tourist) factors, such as personal motivators or past experience, and "external" factors, e.g., advices, information about the products, or the climate of the destination [18]. Context-aware recommender systems can represent and deal with these influencing factors by extending the traditional two-dimensional user/item model that relies only on the ratings given by a community of users to a catalogue of items. This is achieved by augmenting the collected ratings with data about the context of an item consumption and rating [1]. For example, there are places of interest (POIs) that may be liked only if visited on summer (or winter). If the system stores, together with the

M. Hepp and Y. Hoffner (Eds.): EC-Web 2014, LNBIP 188, pp. 77–88, 2014.

rating, the situation in which a POI was experienced, it can then use this information to provide more appropriate recommendations in the various future target contextual situations of the user.

The first challenge for generating context-aware recommendations is how to identify the contextual factors (e.g., weather) that are truly influencing the ratings and hence are worth considering [3]. Secondly, acquiring a representative set of in-context ratings (i.e., ratings under various contextual conditions) is clearly more difficult than acquiring context-free ratings. Finally, extending traditional recommender systems to really exploit the additional information brought by in-context ratings, i.e., building a more effective and useful service, is the third challenge for context-aware recommender systems.

In this paper, we focus on the last challenge and we present a concrete mobile context-aware recommender system called STS (South Tyrol Suggests) that is available on Google Play Store. STS recommends places of interest (POIs) in South Tyrol (Italy) by exploiting various contextual factors (e.g., weather, time of day, day of week, location, mood) and an extended matrix factorization rating prediction model. STS can generate recommendations adapted to the current contextual situation, for example, by recommending indoor POIs (e.g., museums, churches, castles) on bad weather conditions and outdoor POIs (e.g., lakes, mountain excursions, scenic walks) on good weather conditions. The user's preference model is learned using two different sources of knowledge: *personality*, in terms of the the Five Factor Model, that the system acquires with a simple questionnaire, and *in context ratings* that the system actively collects from the user. Exploiting the user personality STS can personalize rating requests and recommendations even for new users (cold start). This novel feature for context-aware recommender systems is supported by the fact that user personality is known to be correlated with user tastes and interests [16].

In previous articles we assessed the STS recommendation algorithm and active learning performance by using classical metrics such as Mean Absolute Error and perceived user satisfaction with the recommendations [9,6,5]. In this article we report the results of the system usability in a controlled live user study. Moreover, we have analysed the log data of the system interactions with more than 500 users that have freely downloaded and tried STS through Google Play Store. The outcome is that users largely accept to follow the supported human-computer interaction and find the user interface clear, user-friendly and easy to use. Moreover, we describe here the user feedback, which gives us a valuable indication for future system improvement.

2 Related Work

Adomavicius et al. [1] have identified three context-aware recommendation models: contextual pre-filtering, contextual post-filtering, and contextual modelling. Contextual pre-filtering (or contextualisation of recommendation input) uses information about the current context for selecting the relevant set of rating data and then predicts ratings using any traditional two-dimensional recommendation technique. For instance, one recent example of contextual pre-filtering is

Semantic Pre-Filtering (SPF) proposed by Codina et al. [8]. It exploits a local Matrix Factorization (MF) model trained on the ratings acquired in contextual situations that are identical or influencing the ratings similarly to the target contextual situation.

In contextual post-filtering (or contextualisation of recommendation output) instead, after predicting ratings using any traditional two-dimensional recommender system trained on the entire data set, contextual information is used to adjust the resulting recommendations. Filter Post-Filtering (Filter PoF) and Weight Post-Filtering (Weight PoF), proposed by Panniello et al. [14], are two concrete examples of contextual post-filtering. They filter or weight the recommended items based on their relevance to the user in a specific target context.

Finally, in contextual modelling (or contextualisation of recommendation function), contextual information is directly used in the modelling technique as part of the rating prediction. The Context-Aware Matrix Factorization (CAMF) approach exploited in the ReRex iPhone app [3] and the InCarMusic Android app [2] is an example belonging to this category. It extends traditional MF rating prediction techniques by incorporating additional model parameters (i.e., baselines) that model how the rating for a place of interest (POI) (as for ReRex) or music genre (as for InCarMusic) deviates as effect of context.

An important aspect of context-aware recommender systems, especially those operating on mobile devices, is the supported human-computer interaction. In spite of the widely recognised importance of the recommender system user interface, mainstream research has been focusing mostly on the core rating prediction algorithms that are assessed through offline evaluations. Littler emphasis has been done on issues related to the proper design of the human-computer interaction. As an example of the second type of analysis we mention the work of Park et al. [15]. They proposed a context-aware and group-based restaurant recommender system for mobile devices and evaluated its usability using the System Usability Scale (SUS) [7]. That is a ten-item questionnaire based on a five-point Likert scale that measures the user's perceived quality of the GUI. In their evaluation they involved 13 users and obtained a system SUS score of 70.58. This indicates a good level of usability, when considering that a SUS score above 68 is assumed to be above average [17].

In [11] the authors present a case study of a constraint-based recommender system that was integrated into a travel advisory system, called VIBE, for the Warmbad-Villach spa resort in Austria. Also in their analysis the authors evaluated the system usability and the perceived customer utility using SUS. They collected the replies of 55 users and obtained an average total SUS score of 81.5. Based on these findings they concluded that the users liked the VIBE user interface. Moreover, similarly to what we have done, they were able to identify a number of usability problems that they could address in a next system release.

We believe that system usability must play a crucial role in recommender system development, besides the accuracy of the core recommendation algorithm. Analogously to the two previously discussed research works, we have used the SUS questionnaire in order to measure the user's satisfaction with the system.

STS, the system described in this article, has obtained in our experiments a SUS score of 77.92, i.e., well above the system described in [15] and close to that described in [11]. We must observe that only the first system is mobile while the second not, making the comparison of the scores less significant in this second case.

3 Interaction with the STS System

We describe here a typical system-user interaction and illustrate the main system functions. Let us assume that a tourist is looking for a POI to visit near to Bozen - Bolzano, Italy. After the registration to the system (providing birthdate and gender), the system asks the user to fill out the Ten-Item Personality Inventory (TIPI) questionnaire [10], in order to acquire the user's Big Five personality traits (openness, conscientiousness, extroversion, agreeableness, neuroticism) (see Figure 1, left).

The entered birthdate, gender and personality scores are then used by an active learning component [9,5] to identify, and request the user to rate, a small set of POIs. This information is estimated to best improving the quality of the subsequent recommendations (see Figure 1, right). We note that the system generates personalized rating requests, relying neither on explicit (e.g., ratings) nor implicit feedback (e.g., item views), which is usually not available for newly registered users.

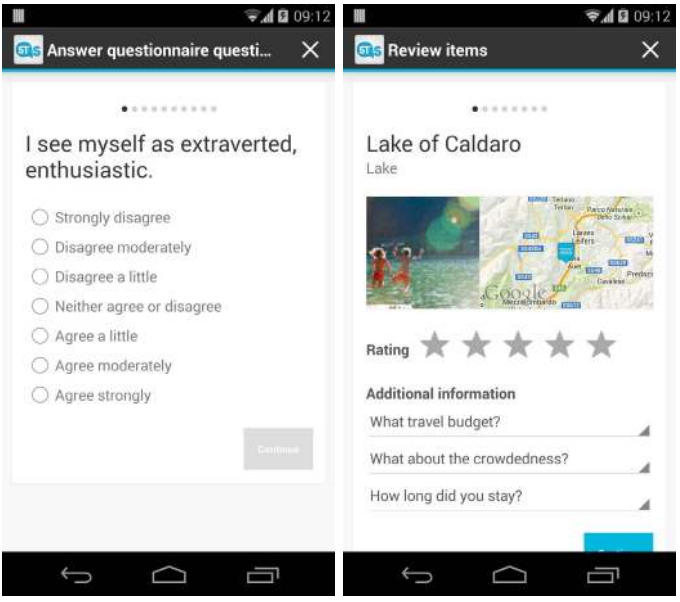

Fig. 1. Active learning

After that preference elicitation phase the system is ready for usage and the user can browse her personalized recommendations through the main application screen (see Figure 2, left). This screen displays a list of four POIs that are considered as highly relevant, considering the current user's and items' contexts. We note that some of these contextual conditions are automatically acquired by the system (e.g., user's distance to the POIs, weather conditions at the POIs), whereas others can be specified by the user through an appropriate system screen (e.g., user's mood and companion), as shown in Figure 2, right.

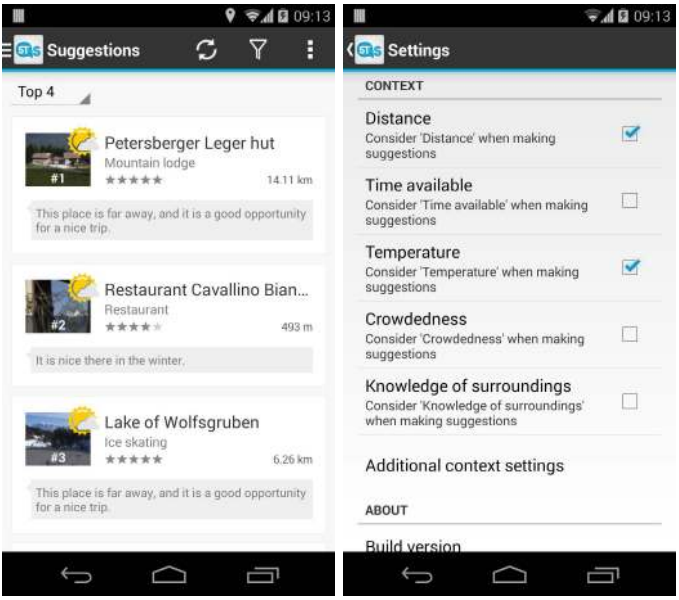

Fig. 2. Context-aware suggestions

If the user is interested in one POI she can click on it and access the POI details window (Figure 3, left). This window presents various information about the POI, such as a photo, its name, a description, its category as well as an explanation of the recommendation based on the most influential contextual condition. Other supported features include the ability to write a review for the POI, to obtain route recommendations to reach the POI (see Figure 3, right) and to bookmark the POI, which then makes it easy and fast to access it later on.

4 Recommendations Computation

STS implements a rich client always-online architecture, i.e., the client has been kept as thin as possible and it works only in a limited way offline. The client application has been developed using the open-source Android platform and

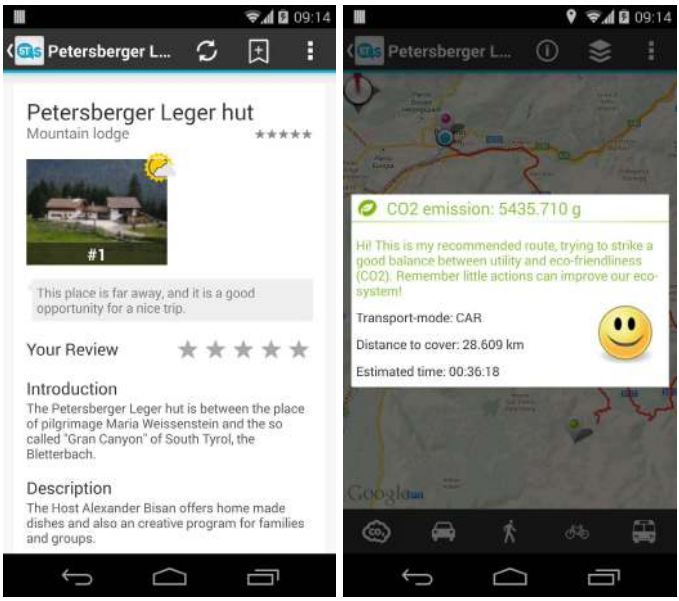

Fig. 3. POI details

implements the presentation layer (GUI and presentation logic). The server application is based on Apache Tomcat server and PostgreSQL database. It implements the data and business logic (recommendation). It makes use of web services and data provided by the Regional Association of South Tyrol's Tourism Organizations (LTS[1]), the Municipality of Bolzano[2] and Mondometeo[3]. These data sources provide descriptions as well as weather forecast information for a total of 27,000 POIs. The server's functionality is exposed via a RESTful web service that accepts and sends JSON objects providing several types of content (suggestions, POIs, reviews/ratings, user profiles).

In order to take into account the current contextual conditions when generating POI recommendations, we have extended the context-aware matrix factorization approach described in [3]. This model, besides the standard parameters (i.e., global average, item bias, user bias and user-item interaction), incorporates baseline parameters for each contextual condition and item pair. Since the original context-aware matrix factorization model fails to provide personalized recommendations for users with no or few ratings (i.e., new user problem), we have enhanced the representation of a user by incorporating user attributes (i.e., age group, gender and the scores for the Big Five personality traits) with a mathematical modelling approach that is analogous to that proposed in [13].

[1] LTS: LTS: `http://www.lts.it`
[2] Municipality of Bolzano: `http://www.comune.bolzano.it`
[3] Mondometeo: `http://www.mondometeo.org`

This allows to model the user preferences even if neither implicit nor explicit feedback is available.

The proposed 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:

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

where q_i, p_u and y_a are the latent factor 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 for contextual condition c_j and item i. Model parameters are learned offline, once every five minutes, by minimizing the associated regularized squared error function through stochastic gradient descent.

5 System Usability Assessment

We have evaluated STS in a user study that involved 30 participants (students, colleagues, working partners and sportspersons) aged between 18-35. The users were asked to look for attractions or events in South Tyrol. The concrete task procedure is as follows: firstly the participants need to consider the contextual conditions that are relevant to them and specify them in the system settings. They were then asked to browse the attractions and events sections and check whether they could find something interesting for them. Also, they were instructed to browse the system recommendations, select one that they believed could fit their preferences and bookmark it. Finally, users needed to fill out a survey and evaluate the system with regard to the perceived recommendation quality and choice satisfaction, whose measurements are adopted from [12].

The rating prediction accuracy (in terms of Mean Absolute Error-MAE) of our recommendation model as well as the performance of the implemented active learning strategy for eliciting ratings were presented in [6,9,5], with the following conclusions: the recommendation model successfully exploits the weather conditions at POIs and leads to a higher user's perceived recommendation quality and choice satisfaction; and the active learning strategy increases the number of acquired user ratings and the recommendation accuracy in comparison with a state-of-the-art active learning strategy.

Here, we report and discuss the system usability results. Several questionnaires have been proposed for evaluating system usability. We have chosen SUS (System Usability Scale) [7] that has become a standard for such analysis. It has been shown that SUS allows to measure perceived system usability using a small sample population (i.e., 8-12 users) [19]. SUS is composed of 10 statements and users reply on a five points Likert scale ranging from "strongly disagree" (1) to "strongly agree" (5): **Q**1: *I think that I would like to use this system frequently.* **Q**2 : *I found the system unnecessarily complex.* **Q**3: *I thought the system was*

easy to use. **Q**4: *I think that I would need the support of a technical person to be able to use this system.* **Q**5: *I found the various functions in this system were well integrated.* **Q**6: *I thought there was too much inconsistency in this system.* **Q**7: *I would imagine that most people would learn to use this system very quickly.* **Q**8: *I found the system very cumbersome to use.* **Q**9: *I felt very confident using the system.* **Q**10: *I needed to learn a lot of things before I could get going with this system.*

The SUS score is computed by summing the score contributions from each item. Each item's score contribution ranges from 0 to 4. For statements Q1, Q3, Q5, Q7 and Q9 (phrased in an positive way) the score contribution is the scale position (from 1 to 5) minus 1. For statements Q2, Q4, Q6, Q8 and Q10 (phrased in a negative way) the contribution is 5 minus the scale position. Then, the sum of the scores is multiplied by 2.5 to obtain an overall system usability score ranging from 0 to 100. We note that the average SUS score computed in a benchmark of 500 studies is 68 [17]. We considered this as a strong baseline for our system since the systems in the benchmark are not mobile and usability for mobile systems is harder to achieve as it requires to deal with the 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.

6 Evaluation Results

Figure 4(a) shows the SUS score of each test user; all but one of our subjects scored better than the benchmark. Overall, STS has obtained an average (over the 30 users) 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) [4]. The margin of error of this SUS score for a 99% confidence interval is 2.84. Hence, with 99% confidence the true SUS score of STS is between 75.08 and 80.76, hence significantly higher than the benchmark.

In Figure 4(b) the Box-and-Whisker diagram of the scores of the 10 SUS statements is plotted. It shows the medians and the distributions of the scores of the ten SUS statements. One can see that the medians are 3, 3.5, or 4 which is a substantially good result (4 is the max score). In addition, we have computed the average replies for the 10 SUS statements. We have observed that the highest average scores are for Q2, Q4, and Q10. This implies that the users have evaluated STS as not complex. They also believe that they did not need neither technical help, nor a lot of things to learn, to be able to use the system.

On the other hand, the lowest scores are measured for items Q9, Q7, and Q5. This implies that users were not extremely confident with using the system and thought that most of the people may not learn quickly using the system. They also found some of the functions in the system not well integrated. Our explanation for these issues is that the user interface was not clear enough to let users understand the true motivation and behaviour of certain functions. For

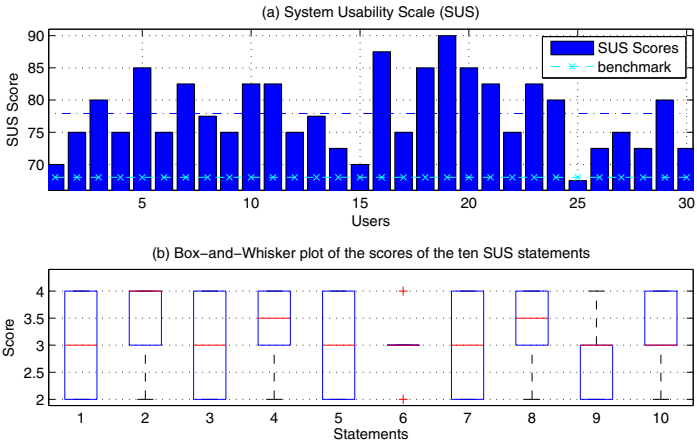

Fig. 4. System Usability Scale (SUS) results

instance, one of our test users mentioned that the personality questionnaire at registration made her mistakenly believe that the app's purpose is to determine her personality type rather than to provide her with relevant POI suggestions based on the current context. We believe that this problem can even worsen if the user is presented with a lengthier questionnaire, which is the reason why we initially decided to use TIPI and not more precise but even more complex approaches.

In order to fix the above-mentioned issues we have improved STS. First of all, we have now replaced the Ten-Item Personality Inventory (TIPI) questionnaire with the Five-Item Personality Inventory (FIPI) questionnaire (see Figure 5, left), which requires less effort. Moreover, we better implemented the active learning process by letting the users to enter their ratings at any moment. The user is presented with a simple and non-invasive in-app notification within the POI suggestions screen informing that better recommendations can be generated if ratings are provided (see Figure 5, right). Finally, we have also improved the user profile page, the instructions, the explanation of the user personality and the presentation of the POI details.

Moreover, in order to better understand the impact of context management on system usability we have compared STS with a similar variant called STS-S. 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). During the experiment, the users were randomly assigned to two groups: one group used STS and the other STS-S. This enabled us to investigate the influence of the incorporation of an important contextual factor, such as the weather, on the usability of the system.

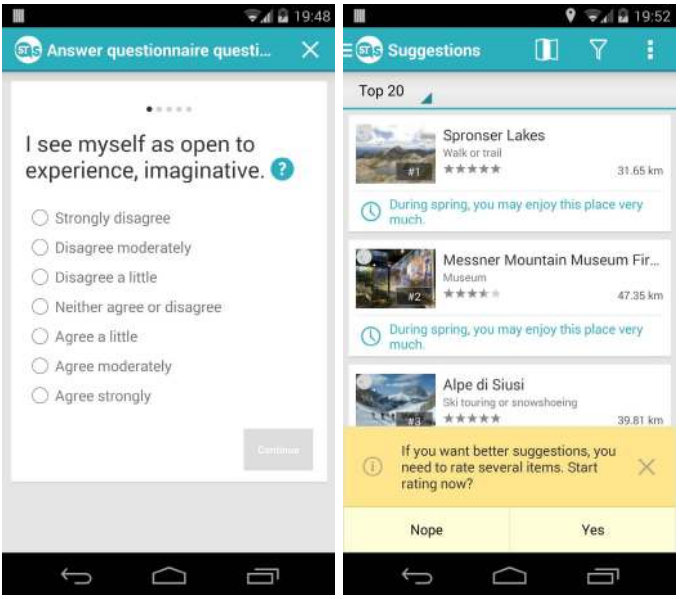

Fig. 5. New user interface design: (left) 5-item personality questionnaire, and, (right) recommendations

STS achieved higher SUS scores compared to STS-S: 78.83 vs 77. Although these two scores are close (and better than the benchmark, i.e., 68) the majority of the users have evaluated STS better than STS-S, in terms of usability. We have computed the t-test, and observed marginal significantly better scores for Q6 and Q10 (see table 1). This indicates that the management of weather forecast data in the proposed mobile context-aware recommender system can increase the system usability in terms of consistency of the system (Q6) and the ability of the users to use the system (Q10). The reason for this is that weather plays an important role in user decision making in tourism application (especially mobile) and also influences the successful adoption of such systems.

Table 1. Comparison of STS and STS-S systems in terms of average scores to SUS statements

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Overall SUS**
STS-S	3.2	3.5	2.8	3.4	2.8	2.8	3.0	3.1	2.8	3.1	77.0
STS	3.0	3.2	3.1	3.3	3.1	3.2	2.8	3.4	2.7	3.4	78.8
p-value	0.27	0.16	0.18	0.40	0.14	**0.08**	0.25	0.19	0.40	**0.11**	0.19

Finally, we would like to note that STS was deployed on Google Play on September 18, 2013, and up to April 6, 2014, 535 users have downloaded and

tried the system. Overall, the system has collected 2,528 ratings and many were entered together with a contextual description of the experience. Among the full set of users, 420 (78.5%) have completed the personality questionnaire and 350 (65.42%) went through the active learning phase. This shows that users largely accept to complete the personality questionnaire as well as the active learning phase to obtain better subsequent recommendations.

7 Conclusions and Future Work

In this paper, we have presented a novel mobile context-aware recommender system named STS, which recommends POIs using a set of contextual factors, such as the weather conditions, the time of day, user's location and user's mood. The novelty of our system resides in several aspects that, we believe, have resulted in the high usability score given by the users. First of all, STS learns to predict users' preferences not only using their past ratings, but also exploiting their personality, which is acquired by asking them to complete a brief and entertaining questionnaire as part of the registration process. Second, the user's personality information has been subsequently used for actively acquiring ratings for POIs that the user is likely to have experienced, and ultimately for producing better recommendations for POIs.

We have conducted a live user study where we measured the system's usability. The results of our user study show that STS has a usability score well above standard benchmarks. Its interface was considered simple and intuitive, and no major usability problems were found during the user study. The main limitation of STS was that not enough clearly it lets the users to understand the true motivation and behaviour of certain functions (e.g. the personality test). We addressed this issue by revising the interaction design, whose benefits will be evaluated in a future work, together with other improvements mainly related the the proactive behaviour of the system. Moreover, in the future we would like to extend the used set of contextual factors by taking into account other dimensions, such as the parking availability and the traffic conditions. We are also currently working on a novel explanation mechanism, that exploits the most influential contextual factor for a given POI rating prediction, to justify why the POI is recommended. We believe that this function can even further improve the usability of the system.

References

1. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-aware recommender systems. AI Magazine 32(3), 67–80 (2011)
2. Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K.-H., Schwaiger, R.: InCarMusic: Context-aware music recommendations in a car. In: Huemer, C., Setzer, T. (eds.) EC-Web 2011. LNBIP, vol. 85, pp. 89–100. Springer, Heidelberg (2011)

3. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context relevance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing 16(5), 507–526 (2012)
4. Bangor, A., Kortum, P., Miller, J.: Determining what individual sus scores mean: Adding an adjective rating scale. Journal of Usability Studies 4(3) (2009)
5. Braunhofer, M., Elahi, M., Ge, M., Ricci, F.: Context dependent preference acquisition with personality-based active learning in mobile recommender systems. In: Zaphiris, P., Ioannou, A. (eds.) LCT 2014, Part II. LNCS, vol. 8524, pp. 105–116. Springer, Heidelberg (2014)
6. Braunhofer, M., Elahi, M., Ricci, F., Schievenin, T.: Context-aware points of interest suggestion with dynamic weather data management. In: 21st Conference on Information and Communication Technologies in Tourism, ENTER 2014 (2014)
7. Brooke, J.: Sus: A quick and dirty usability scale. Usability Evaluation in Industry 189, 194 (1996)
8. Codina, V., Ricci, F., Ceccaroni, L.: Exploiting the semantic similarity of contextual situations for pre-filtering recommendation. In: Carberry, S., Weibelzahl, S., Micarelli, A., Semeraro, G. (eds.) UMAP 2013. LNCS, vol. 7899, pp. 165–177. Springer, Heidelberg (2013)
9. Elahi, M., Braunhofer, M., Ricci, F., Tkalcic, M.: Personality-based active learning for collaborative filtering recommender systems. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (eds.) AI*IA 2013. LNCS (LNAI), vol. 8249, pp. 360–371. Springer, Heidelberg (2013)
10. Gosling, S.D., Rentfrow, P.J., Swann Jr., W.B.: A very brief measure of the big-five personality domains. Journal of Research in Personality 37(6), 504–528 (2003)
11. Jannach, D., Zanker, M., Fuchs, M.: Constraint-based recommendation in tourism: A multiperspective case study. Information Technology & Tourism 11(2), 139–155 (2009)
12. Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C.: Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction 22(4-5), 441–504 (2012)
13. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009)
14. Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., Pedone, A.: Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 265–268. ACM (2009)
15. Park, M.-H., Park, H.-S., Cho, S.-B.: Restaurant recommendation for group of people in mobile environments using probabilistic multi-criteria decision making. In: Lee, S., Choo, H., Ha, S., Shin, I.C. (eds.) APCHI 2008. LNCS, vol. 5068, pp. 114–122. Springer, Heidelberg (2008)
16. Rentfrow, P.J., Gosling, S.D.: The do re mi's of everyday life: the structure and personality correlates of music preferences. Journal of Personality and Social Psychology 84(6), 1236 (2003)
17. Sauro, J.: Measuring usability with the system usability scale (sus), `http://www.measuringusability.com/sus.php` (accessed: January 15, 2013)
18. Swarbrooke, J., Horner, S.: Consumer behaviour in tourism. Routledge (2007)
19. Tullis, T.S., Stetson, J.N.: A comparison of questionnaires for assessing website usability. In: Usability Professional Association Conference (2004)