Long-Term and Session-Specific User Preferences in a Mobile Recommender System

Quang Nhat Nguyen
Free University of Bozen-Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
Quang.NhatNguyen@unibz.it

Francesco Ricci
Free University of Bozen-Bolzano
Piazza Domenicani 3, 39100 Bolzano, Italy
fricci@unibz.it

ABSTRACT

User preferences acquisition plays a very important role for recommender systems. In a previous paper, we proposed a critique-based mobile recommendation methodology exploiting both long-term and session-specific user preferences. In this paper, we evaluate the impact on the recommendation accuracy of the two kinds of user preferences. We have ran off-line experiments exploiting the log data recorded in a previous live-user evaluation, and we show here that exploiting both long-term and session-specific preferences results in a better recommendation accuracy than using a single user model component. Moreover, we show that when the simulated user behavior deviates from that dictated by the acquired user model the session-specific preferences are more useful than the long-term ones in predicting user decisions.

Keywords

Mobile recommender systems, critiquing, user model

INTRODUCTION

Recommender systems (RSs) are intelligent tools addressing the information overload problem and suggesting products which best suit the user needs and wants. To address the limitations of mobile devices and the peculiar characteristics of the mobile usage environment **mobile** RSs have been developed [3,7]. A mobile RS should support simple and short interactions and leverage the knowledge about the user collected in previous interactions.

Critique-based RSs [1,2,3,5,7] seem well suited to this scenario as they assist users to find their desired products through simple dialogues, where the system's product recommendations are provided immediately and the user's preferences are acquired through her critiques to the recommended products. A user makes a critique to a recommended product when some feature of the product is unsatisfactory (e.g., "I want a restaurant cheaper than this") or very important (e.g., "I prefer to eat in an air-conditioned room") for her.

In a previous paper [7] we presented our critique-based mobile recommendation methodology. It exploits a critique-based conversational approach and a composite user-query representation, i.e., where both strict logical constraints and weak similarity-based conditions are used. Our approach integrates both long-term and session-specific user preferences and has been implemented in MobyRek, a mobile recommender that supports mobile users in finding their desired travel products (restaurants). The results of a previous live-user evaluation of MobyRek showed that our recommendation methodology is effective in supporting mobile users in their product selection decisions [7]. In [6] we also showed that the composite query representation employed in our recommendation methodology results in a better recommendation quality over a simpler query representation using either logical or similarity query.

In this paper, we discuss the user preferences model employed in MobyRek and present off-line experiments that test how different components of the user model (UM), i.e., that devoted to represent the long-term and session-specific preferences, influence the system recommendation accuracy. These off-line experiments exploit the log data of real recommendation sessions recorded in a previous live-user evaluation (see [2] for a similar experimental approach). The experiment results show that the exploitation of both long-term and session-specific preferences does improve recommendation accuracy compared to the exploitation of just either the long-term or the session-specific UM component. Moreover, we show that session-specific preferences become more important than long-term ones when the simulated user progressively deviates from the behavior observed in the original session.

RECOMMENDATION METHODOLOGY

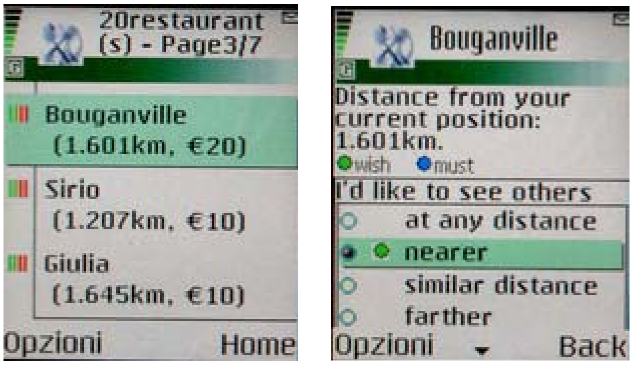
In our approach, a product is represented as a feature vector $x=(x_1,x_2,\ldots,x_n)$, where a feature value x_i can be numeric, nominal, or a set of nominal values.

The user query representation, which encodes the user's preferences, comprises three components, $q=(q_l, p, w)$.

IUI'08, January 13-16, 2008, Maspalomas, Gran Canaria, Spain.
Copyright 2008 ACM 978-1-59593-987-6/ 08/ 0001 $5.00

- The **logical query**, $q_l=(c_1 \wedge c_2 \wedge \ldots \wedge c_m)$, encodes conditions that the recommended products must satisfy. Each constraint in q_l refers to a single feature.
- The **favorite pattern**, $p=(p_1,p_2,\ldots,p_n)$, encodes conditions that the recommended products should match as closely as possible. The conditions in p allow the system to make trade-offs.
- The **feature importance weights** vector, $w=(w_1,w_2,\ldots,w_n)$, models how important a feature is for the user with respect to the others, where $w_i \in [0,1]$ is the importance weight of feature f_i. The system refers to the feature weights when it makes trade-offs or when it searches relaxation for over-constrained (failing) logical queries.

A recommendation session starts when a user asks the system for a product recommendation and ends when the user selects a product or quits the session. A recommendation session evolves in cycles. A recommendation cycle comprises the stage where the recommended products are shown to the user (see Figure 1(a)) and the successive stages where the user browses the products and criticizes one (see Figure 1(b)). When making a critique to a recommended product, the user also assigns the strength (i.e., must or wish) of the expressed preference (see Figure 1(b)). The user's critique is then exploited by the system to compute the new recommendation list that is shown to the user at the next cycle. The details of the recommendation methodology are presented in [6,7].



20restaurant (s) - Page3/7

Bouganville (1.601km, €20)

Sirio (1.207km, €10)

Giulia (1.645km, €10)

Opzioni Home

Bouganville

Distance from your current position: 1.601km.

● wish ● must

I'd like to see others

○ at any distance

● nearer

○ similar distance

○ farther

Opzioni Back

(a) (b)

Figure 1. The MobyRek user interface.
(a)-Recommendation list; (b)-Critique on a numeric feature

USER PREFERENCES MODEL

The system's computation of the recommendation list for a user is based on the current model of the user's preferences. In our approach, the user model incorporates both long-term and session-specific user preferences. Acquiring session-specific preferences helps the system timely capture the changes in the user's preferences, whereas exploiting long-term preferences allows the system to automatically infer some of the user's preferences (see [4] for a discussion on user modeling decomposition).

A user's long-term (stable) preferences (e.g., a preference for non-smoking room) are those inferred from the user's recent recommendation sessions, and remain true throughout these sessions. In our approach, the system infers a user's long-term preferences by exploiting the following knowledge sources (see [7] for more details).

- The user's default preferences that are set by the user and saved in her mobile device's memory.
- The past on-the-go recommendation sessions of the user and other similar sessions.
- The user's pre-travel selection and the past pre-travel selections of other similar users.

In contrast, the user's session-specific preferences (e.g., a desire to eat a pizza) are those transient and specific to a session. Session-specific user preferences cover both contextual and product-feature preferences. Contextual preferences characterize a user's request context whereas product-feature preferences express a user's taste and like. For example, in the restaurant recommendation problem, space-time constraints may indicate that a user wants to see only those restaurants open on some specific day and not too far from the user's position, whereas a product-feature preference may express that the user prefers to eat a pizza. In our approach, the system acquires the user's session-specific preferences in the following ways.

- Space-time constraints are automatically computed by the system exploiting the user's position and time of request.
- The initial preferences that the user explicitly specifies at the beginning of her recommendation session.
- The preferences derived from the user's critiques to the recommended products within the current recommendation session.

In this paper we precisely measure how the two (long-term and session-specific) UM components influence the system recommendation accuracy. In the next section, we shall describe the experiments conducted to answer this research question.

EXPERIMENTAL EVALUATION

We previously tested MobyRek with real users [7], and the log data of this live-user test (consisting of fifteen recommendation sessions) was recorded. In this section, we discuss how we exploited that data in a simulation aimed at evaluating the impact of the two (long-term and session-specific) UM components on the system recommendation accuracy.

We introduced three system variants that exploited different components of the user preferences model. The experiment procedure, followed by all the three system variants, consists of three main steps. First, for each tester's original session the system builds the initial query representation $q^0=(q_l^0,p^0,w^0)$. Second, the system uses the initial query to compute the first recommendation list. Third, the system checks the position of the "target" product (defined soon later) in the first recommendation list. For each system variant, the system computes the average position of the target product over all the simulated original sessions. Finally, the three system variants are compared with respect to the quality of the first-cycle recommendation lists, i.e., the average position of the target product in the first recommendation lists.

- **sysMR**. This variant exploits both the long-term and session-specific user preferences. In particular, the initial logical query q_l^0 is built exploiting the user's space-time constraints and explicit initial preferences stated as must; the initial favorite pattern p^0 is built integrating the user's long-term preferences pattern and explicit initial preferences stated as wish; the initial feature weights w^0 are computed exploiting the history of the user's critiques.
- **sysLT**. This variant exploits only the long-term user preferences. In particular, this variant does not employ the logical query component; p^0 is the user's long-term preferences pattern; and w^0 are computed exploiting the history of the user's critiques.
- **sysSS**. This variant exploits only the session-specific user preferences. In particular, q^0 is built exploiting the user's space-time constraints and explicit initial preferences stated as must; p^0 is built exploiting the user's explicit initial preferences stated as wish; w^0 are equal weights.

We performed two experiments. In the first one, the testers' recommendation sessions (i.e., recorded in the live-user test) were replayed in an incremental way. Here, "replayed" means that the initial query made by the user in the real interactions was recomputed, considering or not certain UM components. In the second experiment, for each simulated user her long-term UM was enriched exploiting the critiques and the selected product of the original session. Hence, we would like to see if the results obtained in the two experiments, before and after the long-term UM is enriched, are different, i.e., if using an enriched long-term UM results in some improvement in the quality of the first recommendation lists.

At the third step of the experiment procedure, the system checks the position of the "target" product in the first recommendation list. Here, the "target" product is that the simulated user is supposed to select in the replayed session. We consider two alternative definitions of the target product.

- The target product is exactly **the one that the user selected** in her original session.
- The target product is a product **similar to the one selected** by the user in her original session.

In the second case we rank all the products in the catalogue according to their similarity to the (real) selected product, and take as target a similar one. The motivation of this is to measure how the performance of the three systems varies when the simulated target product is not exactly, but similar to, the real selected one, i.e., the simulated user deviates from the real behavior. We expect that as the target product is farther from the selected one the importance of the long-term UM component decreases.

In both experiments the same products catalogue, consisting of 84 items, was used. For the *sysLT* variant, the system exploits only the user's long-term preferences to rank the products (according to their similarity to the (*p*,*w*) pair) and, hence, for *sysLT* the size of the first-cycle recommendation list is always equal to 84 – the size of the products catalogue. Conversely, *sysMR* and *sysSS* variants first discard those products not satisfying the user's explicit initial constraints, i.e., the initial must conditions. Therefore, for *sysMR* and *sysSS* the average size of the first-cycle recommendation lists is smaller than 84, and actually it was 69.07 on average over the fifteen simulated sessions. In fact, we analyzed the log data of the testers' recommendation sessions to understand why this average size was rather large, and realized that, though all the testers had explicitly indicated some initial must conditions, most of the products satisfy these indicated constraints.

The First Experiment Results

In the first experiment, for each replayed session the target product was exactly the one selected in the original session. The performance of the three system variants is shown in Figure 2 with the "all sessions" label.

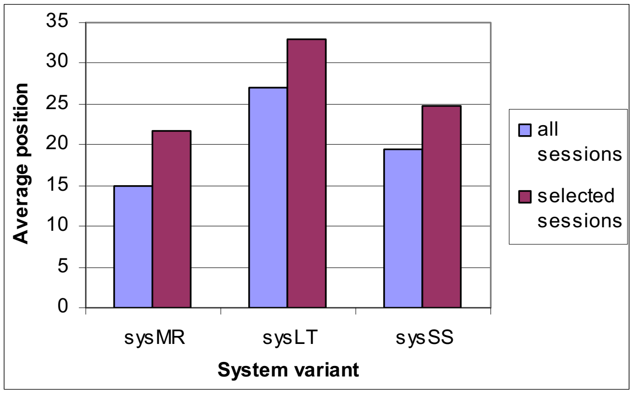


Figure 2. The average position of the selected product in the first-cycle recommendation lists.

As shown in Figure 2, the *sysMR* variant, which exploited both UM components, outperformed the other two variants that exploited only one UM component. In particular, the average position of the selected product in the first recommendation list was 15, 27, and 19.53 for *sysMR*, *sysLT*, and *sysSS*, respectively (the lower value the higher position, as 1 means the first position in the list). The average position of the selected product in *sysMR* was 44.44% and 23.19% higher than those in *sysLT* and *sysSS*, respectively. Also in Figure 2, we can observe that the average position of the selected product in *sysSS* was (27.67%) higher than that in *sysLT*, which means that the session-specific UM component played a more important role than the long-term one.

One may argue that the comparison was biased in favor of *sysMR*, since the experimental data was collected in the live-user test of MobyRek that actually used *sysMR*.

Therefore, one can hypothesize that an item was selected because it appeared in the first screen of the first recommendation list. Hence, to provide an unbiased comparison, we performed the same comparison as before but discarding, in the computation of the average position of the selected product, the simulated sessions where the selected product appeared in the first screen (i.e., within top-3). In other words, we assume that the user did neither browse the full recommendation list nor see the selected product in the first recommendation list. The comparison of the performance of the three system variants on this subset of simulated sessions is shown in Figure 2 with the "selected sessions" label. Even in this case, *sysMR* produced the best first recommendation list, compared to the other two variants. In particular, the average position of the selected product was 21.7, 33, and 24.8 for *sysMR*, *sysLT*, and *sysSS*, respectively. The average position of the selected product in *sysMR* was 34.24% and 12.5% higher than those in *sysLT* and *sysSS*, respectively.

The Second Experiment Results

The second experiment was aimed at understanding the impact of an enriched long-term UM on the quality of the first recommendation lists. For each simulated user, first the long-term UM was enriched with the knowledge derived from the user's critiques and the selected product in her original session, and then the initial query made by the user in the real interaction was re-executed including or not certain components of the UM.

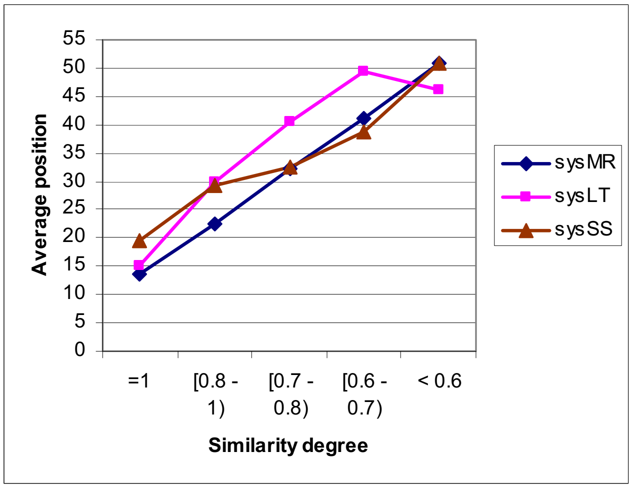


Figure 3. The average position of the target product, after the long-term UM component is enriched.

We first look at the simulations where the target product is exactly the one selected by the user in her original session. In Figure 3 these simulations correspond to the target product with similarity degree equal to 1. In this case, *sysMR* was still the best approach, but *sysLT*, the variant using the long-term UM, was the second, i.e., the situation is different from what observed in the first experiment. This is clearly motivated by the fact that the long-term UM was enriched. Specifically, the average position of the selected product was 13.73 and 14.93 for *sysMR* and *sysLT*, respectively. The result for *sysMR* even improves that obtained in the first experiment.

Let us now consider the results when the target product is only **similar** to the one selected by the user in her original session. Figure 3 shows that *sysMR* still produces a better quality of the first recommendation lists, compared to *sysLT* and *sysSS*. Moreover, for all the values of similarity smaller than 1 and greater than 0.7 the session-specific preferences became more useful than the long-term ones. As the similarity degree decreases further, the quality of the first recommendation lists become poorer for all the three system variants.

CONCLUSION AND FUTURE WORK

In this paper, we have described how long-term and session-specific user preferences are integrated and exploited in a critique-based mobile recommender system. We have shown the benefit of exploiting both UM components with respect to the quality of the first recommendation list.

Our recommendation approach supports different types of users by offering, at start-up, different options for the search initialization. In future, we want to understand how different UM components influence the system recommendation performance for different user types.

REFERENCES

1. Burke, R. Interactive critiquing for catalog navigation in ecommerce. *Artificial Intelligence Review 18*, 3-4 (2002), 245-267.
2. Chen, L. and Pu, P. Preference-based organization interface: Aiding user critiques in recommender systems. In *Proc. 11th Int'l Conf. User Modeling*, Springer (2007), 77-86.
3. Kim, C.Y., Lee, J.K., Cho, Y.H. and Kim, D.H. Viscors: A visual-content recommender for the mobile Web. *IEEE Intelligent Systems 19*, 6 (2004), 32-39.
4. Kobsa, A. Generic user modeling systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W. (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer-Verlag, 2007, 136-154.
5. McGinty, L. and Smyth, B. Adaptive selection: An analysis of critiquing and preference-based feedback in conversational recommender systems. *Int'l. Journal of Electronic Commerce 11, 2* (2006), 35-57.
6. Nguyen, Q.N. and Ricci, F. Replaying live-user interactions in the off-line evaluation of critique-based mobile recommendations. In *Proc. Recommender Systems 2007*, ACM Press (2007), 81-88.
7. Ricci, F. and Nguyen, Q.N. Acquiring and revising preferences in a critique-based mobile recommender system. *IEEE Intelligent Systems 22*, 3 (2007), 22-29.