Critique-Based Mobile Recommender Systems

Francesco Ricci and Quang Nhat Nguyen

eCommerce and Tourism Research Laboratory, ITC-irst
Via Solteri n.38, Polo Tecnologico, 38100 Trento (TN), ITALY
{ricci,quang}@itc.it

Abstract

Recommender systems provide targeted product suggestions for users either overwhelmed by the large number of alternative options offered nowadays in eCommerce websites, or not having enough knowledge to autonomously select the most suited product. Recommender systems are particularly useful for mobile users; here decisions must be normally taken in a short time and the effort required for interacting with the system must be limited as much as possible. In this article, we propose an approach to the generation of mobile recommendations based on the interactive elicitation of user needs and wants through critiques. In this approach the system does not mandatory require the user to explicitly communicate her preferences at the beginning of the interaction; but rather involves her in a dialogue where the system proposes candidate products, and the user feedbacks her critiques about the recommended products. These critiques are then interpreted and incorporated in the user's preferences model managed by the system. This results, step by step, in a better understanding of the user's preferences and needs and in a better ranking of products. The proposed approach has been implemented in a mobile travel recommender system which aims at supporting on-the-move travelers in the selection of travel related services (restaurant). In this article we present the results of the empirical evaluation of this system.

1. Introduction

E-commerce web sites, such as Shopping.com or Expedia.com, offer nowadays large catalogues of different products. When these catalogues are searched often a potentially overwhelming set of options could be retrieved and the user must be able to filter out irrelevant products, compare options, and select the best one(s). In these situations, recommender systems ([16]) can help the user to take a decision suggesting those products which best suit his needs and preferences.

User preference elicitation is a fundamental step in every recommender system. Many approaches have been proposed to acquire user preferences and build a reliable user model. In the most straightforward one, user preferences are collected by explicitly asking the user [2]. Preferences acquired in this way typically represent a subset of the full user's "needs and wants". In fact, users usually do not like to input lots of (personal) data and may also have difficulties in understanding the questions and the rationale of some system requests. In other cases is the system designer that limits the number of preferences to those required by the recommender algorithm. Hence, usually few preferences can be acquired, and, in those systems based on query processing, such as content-based ones ([5]), the query encoding these preferences may return a large result set (product list). This approach has some additional disadvantages. First, the user must have enough knowledge about the problem domain (e.g. digital cameras rather than mp3 players) to be able to formulate a meaningful query. Second, user's preferences may be uncertain or incomplete initially, but the user can better understand what he really wants as he interacts with the system and acquires new information. Third, many users are reluctant in revealing their preferences until they get some real benefit from the system. Finally, the required effort for the input of a complete query may be excessive; especially for a mobile user who accesses the catalogue through a mobile phone or a PDA.

To cope with these problems some researchers have proposed methodologies for acquiring users' preferences through the analysis of the user's navigation behavior [13], [3], [6]. Despite the clear advantage of these approaches, i.e., that of requiring lower user effort, they have two important shortcomings. First, user's actions need to be interpreted and then translated in user preferences. Understanding the meaning of user's actions requires a reference behavioral model of the product search process that relies on specific domain knowledge. Second, implicitly collected preferences are noisy since the reason and objective of a (user) action vary from person to person, and from context to context, and cannot be determined with certainty only observing the human-computer interaction.

Another approach, which has recently received much interest, is eliciting user preferences through a more diversified human-computer interaction, i.e., not based uniquely on querying the user about her preferences. In conversational recommender systems, for instance, at each interaction cycle, the system can either *ask* the user a preference or *propose* a product to the user. The user can reply either by *answering* to the question posed or by *criticizing* the system proposal. Conversational recommender systems may implement only the *asking/answering* conversation mode [9], only the *proposing/criticizing* conversation mode [4], [15], or both [18]. It is worth noting that a conversational system can interact with users either through typing/clicking or with a natural language interface (e.g., [18]). Our discussion in this article is limited to the first interaction modality.

Although many successful web-based recommender systems exist, only few have been designed for mobile users (e.g., [6], [17], [19], [3]). Moreover, very few existing mobile recommender systems are conversational; and most of them run only on PDAs (i.e., Palm or Pocket PC), and they are not appropriate for the much more popular mobile phones. Mobile phones have smaller screens, limited keypads, and supporting a conversation on such a device is extremely difficult. Designing a mobile recommender system requires the proposed recommendation methodology to overcome the limitations of the mobile usage environment and be suitable to mobile users' behavior.

In this article we briefly illustrate a mobile recommender system called MobyRek that exploits the *recommendation by proposal and critiquing* techniques. The system has been designed to run on a mobile phone with limited input from the user. The system search functionality lets the user to formulate both must and wish conditions, and returns ranked list of products. The result of an empirical evaluation of the system shows that it can effectively support product selection processes in a user friendly manner.

2. Recommendation Methodology

In our approach, products are represented as feature vectors in an *n*-dimensional space For example, a restaurant may be represented in the six dimensional vector space: *X=<Name,Type,Location,AverageCost,OpeningDays,Characteristics>*, and the vector *("La-Berta", {pizzeria}, "Trento", 20, {7,1}, {parking,smoking_room})* represents a restaurant, named "LaBerta", whose type is pizzeria, location is in Trento, average cost is 20, opening days are Saturday and Sunday, and with parking and smoking rooms.

In our approach, a user's preferences model incorporates both long-term and session-specific preferences of that user. Given a user's request, the system exploits the user's preferences model to initialize the search query which consists of the three following components.

- **The logical query** (Q_L) models must conditions that need to be independently satisfied by any of the recommended products. The logical query is a conjunction of logical conditions (constraint) on single product features.
- **The favorite pattern** (*p*) models wish conditions that should be matched as many as possible. Wish conditions allow trade-offs to be made, i.e., a good match on a condition could compensate a bad match on another one. The favorite pattern is represented as a vector in the same vector space of the product representation.
- **The feature importance weights** vector (*w*) is again *n*-dimensional and models how much each feature is important with respect to the others.

For instance, the query $<Q_L=(x_3=Trento)\wedge(x5\supseteq\{7,1\}),\ p=(?,\{spaghetteria\},?,?,?,?),\ w=(0,0.6,0,0.4,0,0)>$ models a user who wants a restaurant in Trento, open on Saturday and Sunday, and he prefers spaghetti restaurants to others. The user considers the restaurant type as the most important attribute and then the cost, whereas the other attributes are not important.

A recommendation session starts when a mobile user asks for some particular product; and ends when either the user selects a product or hangs up without selecting anything. A recommendation session evolves in cycles. At each cycle some recommended products are shown to the user. Figure 1 illustrates the logical flow of recommendation process.

Given a user's initial request, such as "let me see a good restaurant for tonight", the system initializes the user's preferences model by exploiting multiple knowledge sources of user-related data. In particular, the user's query is initialized by the system exploiting: a) the user's current space-time information, b) the user's previous selections, c) the past selections of a community of users, and d) the user's explicit initial preferences. The detailed algorithm used by the system to build the initial user query is described in [10]. The system uses this initial representation to compute the first recommendation list. The system firstly filters out those products which do not satisfy the logical query Q_L; and then ranks those satisfying Q_L according to their similarity to the $<p,w>$ pair (i.e., the favorite pattern and the feature weights vector, respectively) [12]. If no products in the catalogue satisfy the initial logical query Q_L, the system finds a minimum number of constraints that if discarded from Q_L make the query satisfiable. In this relaxation, a constraint involving a less important feature is considered (i.e., for relaxation) before another involving a more important one. The relaxed constraints are then converted to wish conditions, and incorporated in the favorite pattern *p* (see also [7], [4] for similar approaches). Only the *k* best products in the final ranked list (i.e., those which satisfy Q_L and are most similar to $<p,w>$) are shown to the user at the current cycle.

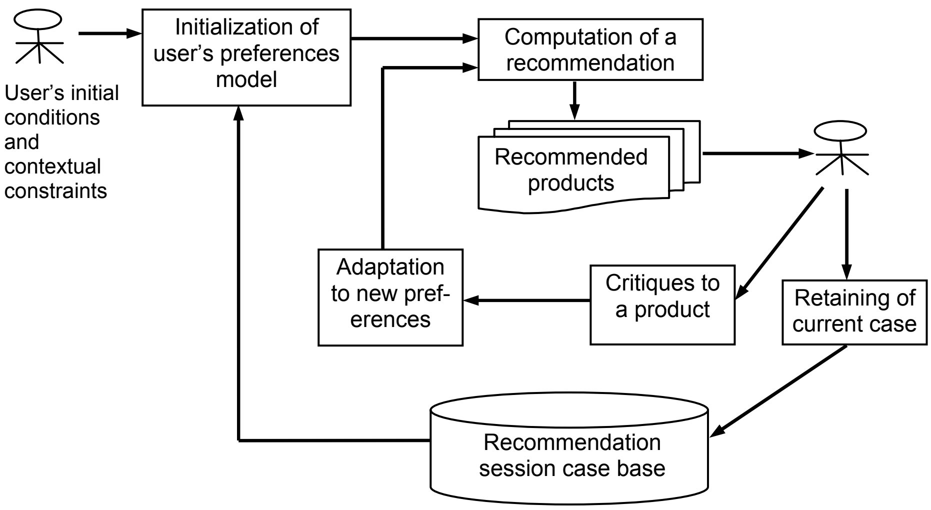
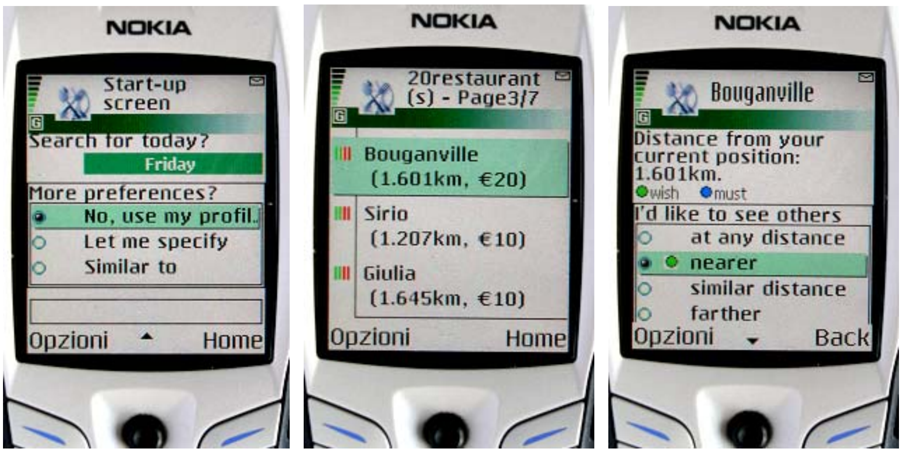


Figure 1. On-the-move recommendation model.

After the system has computed and displayed a list of recommended products, three situations may occur. The user may choose one recommended product and terminate the recommendation session successfully. In a second situation, the user may be somewhat interested in one of the suggested products, but one (or more) of the product features may not satisfy him. In this case the user can criticize the product and better specify his preferences regarding these unsatisfactory features. Moreover the user may state if these additional preferences are "must" or "wish" conditions. Must conditions modify the logical query Q_L while wish ones change the favorite pattern p. For example, a pedestrian user may criticize that he wish to see other similar but nearer products (see Figure 2-c). Such critiques are used by the system to adapt the current query, and to produce a new recommendation list which incorporates the newly elicited preferences (see [11], for more details about the system's query adaptation to user critiques). In a third situation, none of the products recommended are considered acceptable and the user terminates the session with a failure. When a recommendation session finishes, regardless the fact that the session terminated with a success or a failure, it is retained as a case for future references [1]. In this way, past recommendation sessions can be exploited by the system in the initialization of the user's preferences model [10].

3. Empirical Evaluation

We have developed a system prototype, called MobyRek that implements the proposed recommendation approach. This prototype has been developed using Java2ME MIDP 2.0 and runs on a Nokia 6600 phone equipped with a GPS receiver (for position detection). We have evaluated MobyRek with respect to the functionality, efficiency, and convenience dimensions [14].



(a) (b) (c)

Figure 2. The mobile user interface.

The test procedure was executed by fifteen testers and was structured in three phases. In the first one (*training)* the tester was introduced to the Nokia 6600 phone and to MobyRek functionality and usage. In the second phase (*testing)* the tester was asked first to think about the attributes of the desired restaurant, and then to search one with those characteristics. The tester was asked to add the selected restaurant, if this could be found, to the phone memory. Moreover the tester was asked to rate the selected restaurant with respect to his preferences and needs (1: weakly accepted, 5: perfectly suited). In the third phase the tester was asked to fill a usability and user satisfaction survey [8].

The experiment results are summarized in Table 1. All the testers were able to complete the task and find a restaurant. Furthermore, the testers generally rated the restaurants with high rates (i.e., the average rating was 3.87).

	User															Ave.	Std. dev.
	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12	u13	u14	u15		
Length	3	2	4	3	3	2	2	2	6	3	3	3	3	3	2	2.93	1.03
Rank	1	4	2	2	1	1	3	1	1	3	4	1	1	2	1	1.87	1.13
Rating	3	5	3	5	3	4	3	4	4	4	3	4	5	3	5	3.87	0.83

Table 1. Interaction length, rank and rate of the selected item.

Regarding the interaction length, i.e., the number of recommendation cycles needed to identify a good product, Table 1 shows that the majority of the testers could find their desired restaurant within 2-3 recommendation cycles, and this is quite acceptable. The rank of the selected product, i.e., its position in the final recommendation list, was very high (1.87). This means that the testers exploited and trusted the recommendations provided by the ranked list.

The analysis of the testers' answers to the questionnaire confirmed the results shown in Table 1. In particular, the testers stated that they could effectively and quickly find their desired restaurant using the system. They found the system-user interaction and the GUI pleasant and friendly. All the testers stated that the critiquing function (i.e., used in the system-user dialogues) was very useful in helping them to find their desired restaurant, and was easy to use. In general, all the testers said that they would definitely use the system if available on their mobile phone.

4. Conclusions

In this paper, we have illustrated a critique-based approach to mobile recommendations. This approach mostly collects user preferences through critiques; it assures the reliability of the collected preferences while keeping the user effort relatively low. We have also briefly illustrated a mobile travel recommender system, based on the proposed methodology, aimed at supporting restaurant search. The empirical evaluation of this system has shown that the proposed methodology can effectively support users in finding good options in a fast way. In fact, acquiring user preferences through critiques has some advantages. First, the collected preferences are explicitly given by the user, hence are more reliable than those derived by an analysis of the user's navigation behavior. Second, critique-based elicitation requires minimal user effort. In fact, the critique to a recommended product is done simply by 2-3 button clicks (see Figure 2-c). Finally, compared to the explicit request to provide a general preference, the request to criticize a real product is more convincing, it provides some immediate benefit to the user, and therefore better motivates the users to further reveal his preferences.

In the future we would like to improve the reactiveness of the system to cope with potential changes of user preferences during the recommendation session. In fact, user's preferences could be vague at the time of the initial request, and could change as the user interacts with the system and acquires relevant information. Secondly, the exploitation of both long-term and session-specific preferences requires a better understanding of how each source contributes to the user preferences modeling, and how each source influences the system's reasoning processes.

5. References

[1] A. Aamodt and E. Plaza. *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*. AI Communications, 7(1):39-59, 1994.

[2] L. Ardissono, A. Felfernig, G. Friedrich, A. Goy, D. Jannach, M. Meyer, G. Petrone, R. Schäfer, W. Schuetz, and M. Zanker. *Personalising On-Line Configuration of Products and Services*. In Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02), p225-229, 2002.

[3] D. Billsus, M. J. Pazzani, and J. Chen. *A Learning Agent for Wireless News Access*. In Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI'00), p33-36, 2000.

[4] R. Burke. *Interactive Critiquing for Catalog Navigation in E-Commerce*. Artificial Intelligence Review, 18(3-4):245-267, 2002.

[5] R. Burke. *Hybrid Recommender Systems: Survey and Experiments*. User Modeling and User-Adapted Interaction, 12(4):331-370, 2002.

[6] M. Dunlop, A. Morrison, S. McCallum, P. Ptaskinski, C. Risbey, and F. Stewart. *Focussed Palmtop Information Access through Starfield Displays and Profile Matching*. In Proceedings of Workshop on Mobile and Ubiquitous Information Access, p79-89, 2004.

[7] A. Felfernig and A. Kiener. *Knowledge-based Interactive Selling of Financial Services with FSAdvisor*. In Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference (AAAI'05/IAAI'05), p1475-1482, 2005.

[8] J. R. Lewis. *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*. International Journal of Human Computer Interaction, 7(1):57-78, 1995.

[9] G. Linden, S. Hanks, and N. Lesh. *Interactive Assessment of User Preference Models: The Automated Travel Assistant*. In Proceedings of the 6th International Conference on User Modeling (UM'97), p67-78, 1997.

[10] Q. N. Nguyen and F. Ricci. *User Preferences Initialization and Integration in Critique-Based Mobile Recommender Systems*. In Proceedings of the 5th International Workshop on Artificial Intelligence in Mobile Systems, (AIMS'04), 2004.

[11] Q. N. Nguyen, D. Cavada, and F. Ricci. *On-Tour Interactive Travel Recommendations*. In Proceedings of the 11th International Conference on Information and Communication Technologies in Travel and Tourism (ENTER'04), p259-270, 2004.

[12] Q. N. Nguyen, D. Cavada, and F.Ricci. *Trip@dvice Mobile Extension of a Case-based Travel Recommender System*. In Proceedings of the 2nd International Conference on Mobile Business (mBusiness>2003), 2003.

[13] D. M. Nichols. *Implicit Rating and Filtering*. In Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering, p31-36, 1997.

[14] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publisher, San Francisco, 1993.

[15] P. Pu and L. Chen. *Integrating Tradeoff Support in Product Search Tools for E-Commerce Sites*. In Proceedings of the 6th ACM Conference on Electronic Commerce (EC'05), p269-278, 2005.

[16] P. Resnick and H. R. Varian. *Recommender Systems*. Communications of the ACM, 40(3):56-58, 1997.

[17] M. van Setten, S. Pokraev, and J. Koolwaaij. *Context-Aware Recommendations in the Mobile Tourist Application COMPASS*. In Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'04), p235-244, 2004.

[18] H. Shimazu. *Expertclerk: Navigating Shoppers Buying Process with the Combination of Asking and Proposing*. In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01), p1443-1450, 2001.

[19] H.-W. Tung and V.-W. Soo. *A Personalized Restaurant Recommender Agent for Mobile E-Service*. In Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), p259-262, 2004.