

Understanding Recommender Systems: Experimental Evaluation Challenges

Fabio Del Missier and Francesco Ricci

eCommerce and Tourism Research Laboratory, ITC-IRST,
via Sommarive 18, 38050 Povo, Italy
{delmissier, ricci}@itc.it

Abstract. The paper discusses some significant issues in the empirical evaluation of interactive recommender: the role of experiments, the contingent and constructive nature of users' interaction strategies, and the generalizability of the results. We propose to adopt as the main evaluation goal the construction of a situation-specific account of the user-system behavior, and we suggest applying the context matching approach to cope with the contingencies of the user behavior. To make clear the limitations of high-level single step experimental evaluations, we present a critical analysis of a case study, a pilot evaluation of a travel recommender system (ITR). The examination of this study shows the danger of overlooking the detailed aspects of user-system interaction, and underlines the need to iteratively refine the evaluation hypotheses and design when no detailed model of the user-system behavior is initially available.

1 Understanding Intelligent Interactive Systems

1.1 Current Evaluation Approaches

Several methods are used to evaluate the various components of interactive intelligent system in different development stages. The accuracy and the performance of the algorithms are appraised through analytical approaches and off-line empirical tests and simulations, following the tradition of artificial intelligence and machine learning. The interface components are usually tested through a set of HCI techniques, such as heuristic evaluations, cognitive walkthroughs, and verbal (or video) protocols. Specific system functions and interactive decision aids can be analyzed through laboratory experiments. Finally, the whole system is evaluated through experiments, questionnaires, and clickstream analysis (borrowing methods from behavioral

sciences and HCI). More recently, the set of techniques has been expanded by web experiments (mainly used for web ‘field’ studies) and cognitive modeling (used in early stages for interaction assessment and in later stages to generate quantitative behavioral predictions). Specific methods have been proposed to deal with adaptive systems (for instance, layered evaluation [1]).

The adoption of different evaluation methods in different stages seems to be a useful heuristic [2]. Nonetheless, each method has its own shortcomings. First, some HCI techniques are rather subjective, and can provide only weak indications. Second, setting up well-designed laboratory experiments is quite complex and costly. Third, the off-line tests of the algorithms tell only a part of the story. In some domains, tests with real users could show a quite different picture of the system effectiveness (because of the GUI impact and the user’s behavior). Finally, the results of the experimental evaluations of specific decision aids are not necessarily generalizable to situations in which these aids are used within a real and complex system. Therefore, the divide-and-conquer approach does not necessarily guarantee fair evaluation results.

1.2 The Contingent and Constructive Nature of Interaction Behavior

Embodied and situated cognition research has pointed out how the specific environmental and physical constraints can structure and shape cognitive processes. Lave [3] was able to show that arithmetic reasoning during a purchase behavior in the grocery store can be sharply different than reasoning in arithmetic test tasks, and she indicated how everyday cognition relies on environmental constraints. The importance of relatively fine-grained details of information presentation for information seeking and choice has been demonstrated by the behavioral research on information display both in the laboratory [4] and in a real setting [5]. Among the behavioral decision researchers, a shared view is that preferences are often constructed by the decision-maker during the accomplishment of a task and not merely retrieved from memory [6]. The process of preference construction can be deeply affected by many task and context factors, including the response mode (the way in which the preferences are elicited).

In the field of HCI, the interest for embodied and situated cognition is justified by the observation that user interface design can significantly influence cognition, changing the effort level associated to different strategies. Many factors can affect the cognitive strategies: the information acquisition mode and its cost [7], the implementation cost of the operators [8], the cost of error recovery [9], the explicit support for some type of strategy, the availability of a suited external representation, the perceptual salience of the information, and the relative importance of accuracy

maximization vs. effort minimization goals [10]. HCI research has also highlighted how very simple interactive task (i.e., moving the mouse and pressing a button) can be performed using different microstrategies [11]. The selection of different microstrategies can be influenced in subtle ways by apparently minor changes of the interface design and can produce significant timesavings in routine interaction behaviors. Understanding the user-system interaction is a very complex problem if the system is equipped with some kind of user model. The system interface can be considered as the user's window on the system, and is able to affect her/his representation through what it makes available and the feedback delivered [12]. At the same time, the user interface is the system window on the user, and it affects the system user model via the information gathered and the input collection mode.

1.3 Toward a Situated Approach

Given the complexity of the interactive intelligent systems and the contingent and constructive nature of cognition, it is very difficult to properly evaluate the user-system behavior. This behavior can be affected by many factors, and it is practically infeasible to manipulate each relevant variable in the experiments. Moreover, it cannot be assumed that the user's behavior will be static: any slight change to the interface or to the system can modify the user's interaction and choice strategies.

From our viewpoint, evaluating a system means trying to understand its behavior as the result of a complex interplay between its functions, the users' strategies, and the specific aspects of the interface. Therefore, the main goal of our evaluation efforts should be the definition and test of a detailed situation-specific account of the user-system behavior. This means that our evaluations will have a rather narrow generalization extent, and that we should carry out detailed interaction analyses. Specific evaluation techniques (see sub-section 1.1) should be carefully applied in preliminary tests, before setting up the experimental evaluations of the final system. These early tests can give us some confidence on the proper behavior of some system components, but they do not guarantee that the real system will work properly.

For the experimental evaluation of the real system, we suggest adopting the context matching approach to cope with the contingencies of the user's behavior [6]. For recommender systems, this means that it will be necessary to reproduce the real decision environment in the experimental setting. Therefore, the real system should be tested, with no change in the available information and databases, in the interface, in the support tools, in the algorithms and parameters. Furthermore, to assure external validity, also the evaluation setting and the sample of users should be selected to be representative. In this way, it will be possible to manipulate in a principled way only a few relevant factors, in order to understand their impact on the user-system behavior

in the real decision environment. This approach will allow us to formulate correct predictions for the evaluation extent.

2 Evaluation of an Interactive Case Based Recommender

To make clear the limitations of high-level single step experimental evaluations, we will present a critical analysis of a case study, a pilot evaluation of our Intelligent Travel Recommendation system (henceforth ITR, 13). This preliminary evaluation was carried out to get some general indications on the system performance and interface, given that the system was still in evolution, the GUI was in a prototypical stage and the case base was not very huge. Furthermore, we did not have any empirically supported model to guide us. Therefore, it seems that this basic experiment could be considered as representative of a typical early evaluation study.

The examination of this pilot test shows the danger of overlooking the detailed aspects of user-system interaction, and underlines the need to iteratively refine the evaluation hypothesis and design when no detailed model of the user-system behavior is initially available. We will show that a detailed analysis of the log data, focussed on the user-system interaction, was able to highlight a series of problems and to identify some sub-optimal interaction behaviors. This analysis suggested potential explanations, new hypotheses, and some methodological changes.

2.1 Problem and System Description

The main purposes of recommender systems are to suggest interesting products and to provide information support for consumers' decision processes [14]. These systems are typically embedded in e-commerce web services, and they take into account the user's needs and preferences in order to propose a suited set of products, relying on specific algorithms and on the knowledge already acquired by the system. Recommender system research is therefore mainly focussed on the issues of information overload, lack of knowledge, trade-off optimization, and interaction cost minimization. The two main recommendation approaches are collaborative-filtering and content-based recommendation [15].

We have designed a novel hybrid collaborative/content-based method, grounded on a case-based approach, which tries to overcome the limitations of the existing technologies. The cases stored by the system are whole user-recommender interactions, comprising the information provided by the user during a specific session, the selected products, and some personal preferences (for registered users only). We developed a recommender prototype for the leisure travel domain [13] that

is built on our method. The goal of the system is to suggest to the user different kinds of travel components (locations, accommodations, attractions, activities, and events) and complete travel packages. The user interacts with the system by specifying queries through a form-based interface, and the system tries to satisfy the user's needs and preferences, helping her/him to find appropriate items in some product catalogues. The system support can be delivered in two interaction stages: the query formulation and the presentation of the results.

During the interaction, the user provides both content and collaborative features. Content features are product properties used to query the product databases, and collaborative features are more general travel preferences. For instance, the hotel rating is a content feature that can be used in an accommodation query, while the preferred transportation means is a collaborative feature.

The basic interaction sequence can be outlined as follows:

- The user specifies a set of collaborative features for the travel.
- The user composes a query using the content features.
- If the query is going to fail (no results will be retrieved from the database), the recommender suggests to relax some specific constraints (relaxation); if, on the contrary, the query is going to produce too many results, the system suggests to tighten it (tightening). The relaxation and tightening functions follow a conversational and mixed initiative model [16].
- The results obtained from the query are sorted according to a specific similarity measure. First the recommender finds a small set of similar interaction sessions, using all the collaborative features specified by the user. Then, it computes a double similarity score for each query result, by taking the product of the case similarity and the item similarity. The case similarity takes into account the current case and the retrieved cases, and is calculated using only the collaborative features. The item similarity (i.e., the similarity between the result item and the items of the same type contained in a retrieved case) is instead computed on the content features. The result items are finally presented to the user in decreasing similarity order. Therefore the sorting function supports the adaptive presentation of the results, a typical feature of adaptive hypermedia systems.

It should also be remarked that a ordinary interaction session will be composed of several queries on different kinds of products, and that, in many cases, a given query will be iteratively refined. Furthermore, it is also possible for the user to request complete travel recommendations; in this case the system will propose some complete travel packages that have been previously assembled by similar users.

The aims of our user-system interaction design are (a) to maximize the match between the user preferences and the features of the available products (via relaxation), and (b) to reduce the information overload (via tightening). Furthermore,

the system intervention is devised to be quite unobtrusive, and the user is always allowed to ignore the suggested query refinement.

2.2 Empirical Evaluation

The empirical evaluation of the system prototype followed a two-group between-subjects design. Two experimental groups were associated with two variants of the recommender system: ITR+ and ITR-. ITR+ is the full functionality recommender, while ITR- is a baseline version, without the interactive query management and the ranking functions. Thus ITR- is not able to suggest how to change a query, and it does not provide any smart sorting of the selected products: the result order mirrors the order of the items in the product catalogue. The recommendations computed by ITR+ were based on a small case base (35 cases), generated by a set of expert and naive users. In both the system variants each result page displays three items.

We have randomly assigned participants to the ITR- ($n=19$) and ITR+ ($n=16$) groups, without mentioning that we were testing two different systems. The users were students and employees of the University of Trento. Their task was to plan a vacation in Trentino, selecting a set of travel products and putting them in a virtual repository (travel bag). The participants were given an explanation of the task, but they did not receive any kind of advice from the experimenter; only the instructions on the web site were available. Before solving the task, they had some time to acquaint themselves with the system (about 10 min). At the end of the experimental session the participants were required to fill in a tailor-made evaluation questionnaire and they were invited to provide free comments and observations.

Hypotheses and Basic Results. We hypothesized that the ITR+ system is able to provide useful recommendation (H1) and to improve search and decision efficiency (H2). We compared the two groups on some theoretically relevant dependent measures, using t-tests with separate variance estimation and Mann-Whitney U tests and obtaining the same pattern of results.

The input information provided by the two groups was essentially equal. The user specified a similar number of collaborative features for each session (ITR+: $M=11.5$, $SD=2.1$; ITR-: $M=12.3$, $SD=1.4$). The mean number of conditions filled in the queries was also very similar (ITR+: $M=4.4$, $SD=1.1$; ITR-: $M=4.8$, $SD=1.2$).

ITR- users formulated more queries than ITR+ users, but this difference did not reach statistical significance (means: ITR+=13.4, ITR-=20.1). The Levene test highlighted a significant difference between the two systems in the variance of the number of queries ($F(1,33)=4.29$, $p<.01$; ITR-: $SD=19.17$; ITR+: $SD=9.25$). ITR- outputs a higher number of results than ITR+ (means: ITR+=9.8, ITR-=42;

$t(33)=2.05, p<.05$). The difference in the number of pages displayed by the two systems is in the same direction, but it is not significant (means: ITR+=71.3, ITR-=93.3). The variance of the mean number of results is again significantly lower in ITR+ than in ITR- (Levene test: $F(1,33)=18.39, p<.0001$). It is interesting to note that the session duration is not different in the two conditions (means: ITR+=31 min, ITR-=28.5 min); this shows that ITR+ users have devoted more time than ITR- users to examine the information content rather than formulating queries or displaying pages. From the result pattern we can conclude that hypothesis H2 is only partially supported and that the two systems are associated to different interaction behaviors.

The number of items collected in the travel bag is similar in the two systems (means: ITR+=4.1, ITR-=5.8), but the mean position of these items is significantly nearer to the top in ITR+ (means: ITR+=2.2, ITR-=3.2; $t(141)=1.96, p=.05$ with log transformed position data). The last result is compatible with H1: despite an approximately equal session time and a lower number of pages and results to examine, the ITR+ participants selected items with high recommendation ranks.

Interaction with ITR+ Recommendation Functions. We gained some support for the ITR+ efficiency and recommendation capability, but there was some indication that the users were encountering some difficulty in the use of the system. In particular, we were warned by the results of a questionnaire item. Thus, we analyzed the log data (excluding the queries with missing or incomplete information), taking into account 216 queries. A diagrammatic representation of the interaction paths is presented in figure 1.

In 19 queries (9%) the users asked the system to provide a complete travel recommendation (a complete travel package), but the system suggestion was added to the travel bag in only 3 cases. The difference between the proportions of selected items after a single item search and after a complete travel request is significant (single item=.42, complete travel=.16, $p<.05$). This result and some of the users' post-test observations highlight a potential system problem, and allow us to formulate two hypotheses: (a) the interface and the interaction for adding complete travels to the bag are not properly designed, and (b) the complete travel recommendations are unsatisfying. Testing these hypotheses will require further experimental work.

197 queries (95% confidence interval for the .91 proportion: .87 to .94) were single item searches: the user looked for a location, an accommodation, or some kind of event or activity. The single item search was repeated in an iterative way until the travel plan was deemed complete. In 65% of the single item queries the system proposed to relax or to tighten the query constraints. These queries were often too specific (74%). In this case, the users could either (a) accept the system suggestion (automatic relaxation), (b) accept the suggestion but modify the query 'by hand', or

(c) try to compose a different query. In the majority of relaxation interactions (69%) the user accepted the system hint. When the user did not find a suited item (queries with no results), in about 31% of cases the system was able to come up with a suggestion that subsequently lead to the selection of a travel item. Thus, it seems that the relaxation hints were accepted by the users and were useful: following the system advice yielded a significantly higher proportion of items added to the travel bag than modifying the query in a different way (suggestion followed=.45; query modified=.21, $p<.05$). However, the presence of a certain number of relaxations 'by hand' casts some doubt on the effectiveness of the interface design.

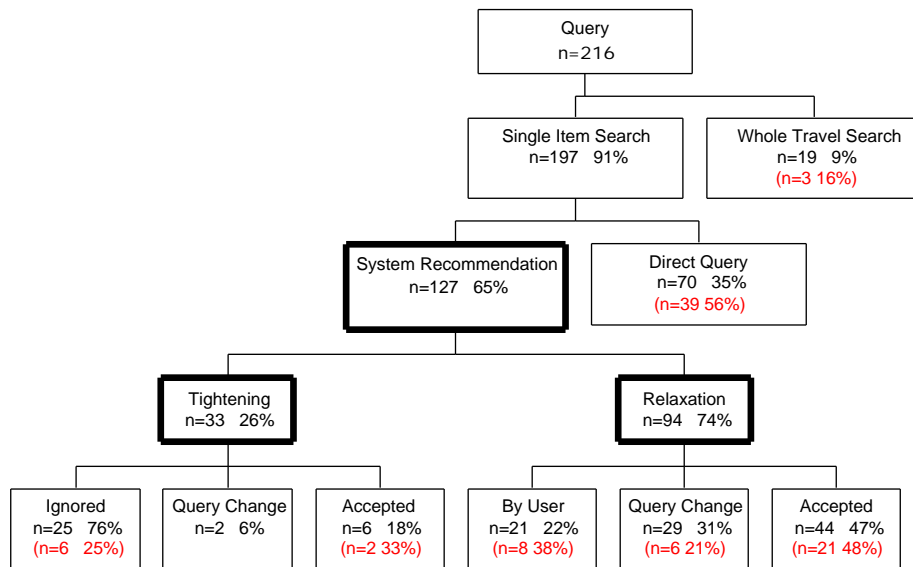


Figure 1: The interaction paths. The boxes with bold borders represent the system intervention. The frequencies and percentages in parentheses describe the users' selections, computed in reference to the action specified in the containing box.

A quite different story can be told about the tightening support functions. In the majority of cases (95% confidence interval for the .76 proportion: .59 to .88), the users ignored the suggestion and preferred to be able to freely explore the whole set of results, even when the number of results was greater than ten. It seems that participants were reluctant to constrain their search, and we can formulate many hypotheses on the reasons of this behavior. Among these we can mention the unwillingness to winnow out potentially appropriate options, the need to explore more deeply an unknown option space, or the cognitive effort associated with the constraining operations. Again, only further theoretically-grounded experiments can help us to obtain a convincing explanation.

3 Why this is not enough? Conclusions and Future Work

Some preliminary indications have been obtained from the pilot study:

- We had some weak indications that ITR+ seems to be able to provide useful recommendations, but the real effectiveness of the decision support needs to be investigated further.
- ITR+ seems to change sharply the users' information-seeking patterns.
- The relaxation support is useful and accepted, even if the interface design should probably be modified. The tightening support and the complete travel recommendations are rarely used, and they should be reconsidered.

Nonetheless, we are far from having defined a detailed user-system account. Our future work will follow the approach described in the previous sections. A set of preliminary test has been planned, to provide independent evaluations of some system components. First, we will evaluate the interface design of the relaxation, tightening, and complete travel recommendation functions in a specific experiment. Simulations will appraise the effectiveness and efficiency of the interactive query management functions. Following the layered evaluation framework [1], a choice experiment will contrast the case-based approach with a utility-based approach (and with a random policy) on their predictive validity; no adaptive sorting will be enabled, and the presentation order will be controlled. These tests will eventually lead to some changes to the interface or to the system.

Finally, we will design a more informative experimental evaluation of the revised system, following the context matching approach. The study will allow us to test some detailed hypothesis and to sketch a preliminary situation-specific model of the user-system behavior. Many changes will affect the design, the measured variables, and the evaluation tools (questionnaires, log protocol). The main planned changes are summarized as follows:

- We will test a detailed set of hypotheses involving: (a) interaction efficiency, (b) recommendation quality, (c) decision and plan quality, (d) user satisfaction, and (e) user knowledge of the target touristic area.
- Case-base features (number of cases and case properties) will be included in the experimental design as factors, to test the impact of the system user model on the case-based sorting.
- Different types of sorting functions (random, case-based, utility-based) will be contrasted on a set of dependent variables (selection time, item position on the result list, ratings of selected items). Furthermore, all the information used for the adaptive sorting (features, similarities, utilities, etc.) will be logged for manipulation checks.

References

1. Karagiannidis, C., & Sampson, D. (2000). Layered evaluation of adaptive applications and services. In P. Brusilovsky, O. Stock & C. Strapparava (Eds.), *Adaptive hypermedia and adaptive web-based systems*, Lecture Notes in Computer Science, Vol. 1892, 343-346, Springer-Verlag.
2. Nielsen, J. (1993). *Usability engineering*. San Francisco: Morgan Kaufmann Publisher.
3. Lave, J. (1988). *Cognition in practice*. Cambridge: Cambridge University Press.
4. Schkade, D. A., & Kleinmuntz, D. N. (1994). Information display and choice processes: Differential effects of organization, form, and sequence. *Organizational Behavior and Human Decision Processes*, 57, 319-337.
5. Russo, J. E. (1977). The value of unit price information *Journal of Marketing Research*, 14, 193-201.
6. Payne, J. W., Bettman, J. R., & Schkade, D. A. (1999). Measuring constructed preferences: Towards a Building Code. *Journal of Risk and Uncertainty*, 19, 243-270.
7. Lohse, G. L., & Johnson, E. J. (1996). A comparison of two process tracing methods for choice tasks. *Organizational Behavior and Human Decision Processes*, 68, 28-43.
8. O'Hara, K. P., & Payne, S. J. (1998). The effects of operator implementation cost on planfulness of problem solving and learning. *Cognitive Psychology*, 35, 34-70.
9. O'Hara, K. P., & Payne, S. J. (1999). Planning and the user interface: The effects of lockout time and error recovery cost. *International Journal of Human-Computer Studies*, 50, 41-59.
10. Fum, D., & Del Missier, F. (2001). Adaptive selection of problem solving strategies, *Proceedings of the twenty-second annual meeting of the Cognitive Science Society* (pp. 313-318). Mahwah, NJ: Erlbaum.
11. Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds Matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied*, 6, 322-335.
12. Fisher, G. (2001). User modeling in human-computer interaction. *User Modeling and User-Adapted Interaction*, 11, 65-86
13. Ricci, F., Arslan, B., Mirzadeh, N., & Venturini, A. (2002). ITR: A case-based travel advisory system. *Proceedings of the seventh European Conference on Case Based Reasoning*, (pp. 613-627). Springer Verlag.
14. Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5, 115-153.
15. Burke, R. (2000). Knowledge-based recommender systems. In A. Kent (ed.), *Encyclopedia of Library and Information Science*, volume 69. New York: Marcel Dekker.
16. Ricci, F., Mirzadeh, N., & Venturini, A. (2002). Intelligent query management in a mediator architecture. *First International IEEE Symposium Intelligent Systems*, Varna, Bulgaria.