Explicit Passive Analysis in Electronic Catalogs

David Portabella Clotet and Martin Rajman

École Polytechnique Fédérale de Lausanne (EPFL)
{david.portabella, martin.rajman}@epfl.ch

Abstract

We consider example-critiquing systems that help people search for their most preferred item in a large catalog. We first analyze how such systems can help users in the framework of three existing example-critiquing approaches (RABBIT, Incremental-critiquing and ATA). Afterwards we consider the use of several novel types of explicit passive analysis to guide the users in their search when their original query returns no hits. We suggest that a user-centric search system together with the right explicit passive analysis makes the users feel more confident in their decision and reduces session time and cognitive effort. Finally we present the results of a pilot study.

Introduction

A Multi-Attribute Decision Problem (MADP) is the problem of finding the best outcome or solution based on the user's preferences. However users may not have an accurate idea of their preferences while searching for a product, especially when the product is beyond the user's domain knowledge. They may begin the search with some vague set of preferences and refine them as they learn more about the different possibilities [1].

An example-critiquing interaction is an iterative process where the user and the system collaborate to find the best solution. Based on the current user's preference model, the system shows a set of candidate solutions and the user gives some feedback to the system so that it can update the preference model. The loop continues until the user is convinced with one of the candidate solutions.

In this poster, we first analyze three example-critiquing systems and then we propose the use of Explicit Passive Analysis to better guide the users in their search. We conclude with the results of a pilot study.

Example-Critiquing systems

One of the first systems to implement such an approach was the RABBIT system [2], where the preference model is a query explicitly given by the user. This system is sometimes referred to as of a *query-building interface*. The candidates shown to the user are simply the list of all items satisfying the query. The query can then be directly reformulated by using options such as ``prohibit'' or “specialize" over an attribute. When the solution space is overconstrained (i.e. there is no solution satisfying the query), the system simply does not show any candidate solution. In contrast, in a *preference-based* approach, the user would have the possibility to define a weight for each constraint and this information would be used by the system to provide a list of ranked partial matches. The RABBIT system is user-centric in the sense that it is the user who guides the interaction, the system just providing the requested information.

The incremental-critiquing approach proposed by McCarthy in [4] shows just 1 candidate solution and a set of compound critiques that describe the feature relationships that exist between the remaining cases. The user has the option of directly updating the query by adding or changing a constraint over an attribute, or by accepting one of the proposed compound critiques. The system selects the 3 best compound critiques by taking the ones with lower support values, as these would eliminate many items from consideration if chosen. This approach has the advantage that, if one of the compound critiques turns out to be interesting to the user, selecting it would probably require less cognitive effort than manually inspecting all the matching items, and extracting and applying the individual critiques one by one.

One important feature proposed in he ATA system [5] and further elaborated in [3] is that the system not only presents the k best ranked items but also a set of alternative suggestions which are significantly different and have the potential to stimulate the expression of hidden preferences and thus avoiding locally optimal solutions. For instance, if the user has not expressed a preference for non-stop flights, presenting an alternative solution that best satisfies the expressed preferences plus the latter one could potentially induce the user to realize that this new preference is important to him.

Explicit Passive Analysis

In the ATA interface passive analysis is provided in the form of alternative solutions. We suggest that providing

the users with an explicit justification that can potentially lead to these alternative solutions will make them feel more confident in their decision process and speed up the interaction. For instance, in the ATA interface the system could inform the user that, given the current preference model, he may potentially be interested in adding a preference about non-stop flights.

We think that a good way to increase user trust is to make users feel that they have reached a good characterization of the domain (at least in their region of interest). Thus, it is not enough that the system presents the best choice to the user, but the user needs to be convinced that it actually is the best one. A natural way to achieve this is that the system makes it easy for the user to discover why there is not a better choice in the database.

In this direction, we propose to use a query-building tool to let users browse and characterize their region of interest and to help them by providing 2 types of explicit passive analysis about their current query, as described hereafter.

During their interaction with the system, users incrementally add constraints, until they eventually end up in an overconstrained situation. Rather than providing partially matching items, we propose to explicitly provide the user with information about the minimal conflicting set of constraints. For instance, in a catalog of second-hand cars, given the current constraints expressed by the user, the system could inform that "*there are no Audi cars & Cheaper than 6000 Euros & Matriculated later than year 2000*" nor "*cars Cheaper than 6000 Euros & with Blue color & Electric windows*". This information can potentially help the user to better understand the solution space.

A second type of explicit passive analysis is to provide concise relaxation suggestions to resolve the minimal conflicting set. While the information about the minimal conflicting set may be useful to characterize the solution space and thus get more confidence, finding which constraints to relax to overcome the situation is not evident for the user and it may involve trying several relaxation combinations. Providing such a list of possible relaxations may lead to a speed up of the interaction. For instance, in the previous example, the system would also inform that there are cars "*if he relaxes the maximum price to 6500 Euros*" or "*if he relaxes the maximum price to 6200 Euros and without Electric windows*".

Pilot Study

We have carried out a pilot study for evaluating our approach. During this pilot study, we have used two binary control variables: (1) whether to show the explicit analysis or not and (2) whether to use a query-building or a preference-based approach. The resulting four interfaces were implemented using the same software so that the "look and feel" remains as similar as possible to get a fair comparison.
As a database, we used a set of 7000 second-hand cars kindly provided by the company Comparis[1], with 35 attributes for each car[2].

We have tuned our prototype to compute the minimal conflicting set for combinations of at most 4 constraints.
Notice that the selection of the appropriate value for this maximum deserves further investigation. Similarly, we have limited to three the maximum number of attributes in a relaxation suggestion.

Although the pilot study involved only 4 users, some interesting preliminary results were obtained:

- Users using the query-building approach felt more control about the system and they appreciated it.
- Users using the preference-based approach did hardly make use of the weights and were dissatisfied because they didn't understand how the items were ranked
- Users using the query-building approach were dissatisfied by the fact that no items were displayed in an overconstrained situation.
- More importantly, users using the explicit passive analysis (in both the query-building and the preference-based approach) were more confident with their choice, while there was no significant difference in the interaction time.

Based on these encouraging results and in order to obtaining more significant results, we are currently preparing a complete user study with 40 users.

References

1. J.W. Payne, J.R. Bettman, and E.j. Johnson. The Adaptive Decision Maker. Cambridge University Press, 1993.
2. M.D. Williams. What Makes RABBIT Run? International Journal of Man-Machine Studies, 21:333-352, 1984.
3. B. Faltings, P. Pu, M. Torrens, and P. Viappiani. Designing Example-Critiquing Interaction. In Proceedings of the International Conference on Intelligent User Interfaces (IRI-2004), pages 22-29. ACM Press, 2004. Funchal, Madeira, Portugal.
4. K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. An analysis of Critique Diversity in Case-Based Recommendation. In proceedings of the Eighteenth International FLAIS Conference (FLAIRS-05). 2005. Clearwater Beach, FL, USA.
5. G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: Te Automated Travel Assistant. In C. Paris A. Jameson and C. Tasso, editors. User Modeling: Proceedings of the Sixth International Conference, pges 67-78.Springer Wien, 1997.

[1] http://www.comparis.ch/
[2] We would like to thank Comparis for sharing their database. An extract can be downloaded from http://icwww.epfl.ch/~portabel/ecatalogs