Implementing Example-based Tools for Preference-based Search

Paolo Viappiani
Artificial Intelligence Laboratory (LIA)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland
paolo.viappiani@epfl.ch

Boi Faltings
Artificial Intelligence Laboratory (LIA)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland
boi.faltings@epfl.ch

ABSTRACT

Preference-based search is the problem of finding an item that matches best with a user's preferences. User studies show that example-based tools for preference-based search can achieve significantly higher accuracy when they are complemented with suggestions chosen to inform users about the available choices. We discuss how suggestions can be efficiently implemented even for large product databases.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors, software psychology*; H.3.5 [**Information storage and retrieval**]: Online Information Services—*Web-based services*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*evaluation/ methodology, graphical user interfaces*

General Terms

Human Factors

Keywords

Personalized search, example critiquing interfaces, flexible query systems, preference elicitation.

1. INTRODUCTION

People frequently use the world-wide web to search through a large collection of items. The most common search facility available on the web is based on a form that is directly mapped to a database query and returns a ranked list of the most suitable options. The user has the option to return to the initial page and change his preferences and then carry out a new search.

In most cases, users do not know exactly what they are looking for. In fact ,psychological studies have shown that people construct their preferences [3] while learning about the available products. Therefore, tools for preference-based search should also help users in formulating accurate preferences. Example-critiquing is such a tool.

ICWE'06, July 11-14, 2006, Palo Alto, California, USA.
ACM 1-59593-352-2/06/0007.

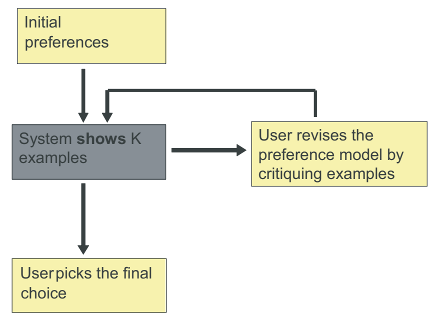

Figure 1: Example-critiquing interaction.

1.1 Example-critiquing

In Example-critiquing, the user is shown examples of options that fit the current preference model well. The idea is that an example either is the most preferred one, or there is some aspect in which it can be improved. Thus, on any of the examples, any attribute can be selected as a basis for critiquing. For instance, if the arrival time is too late, then this can be critiqued. The critique then becomes an additional preference in the model.

Several tools have been proposed in the literature, including the ATA system of Linden et al. [2], SmartClient [4], and more recently dynamic critiquing systems [7]. It has been shown [5] that the elicitation of preferences in this way is effective.

To use a metaphor, the process of example-critiquing is hill-climbing: the user states preferences as long as he perceives it as bringing to a better solution. However, the process might end in a local optimum; a situation in which the user can no longer see potential improvement. For these reasons we display two sets of examples:

- **candidate examples** that are optimal for the preference model, and
- **suggested examples** that are chosen to stimulate the expression of preferences.

The computation of candidates can be done with a *top-k query* [1].

For the suggestions, in [6] we proposed different strategies that use the concept of Pareto-optimality to implement the principle that suggestions should have a high likelihood of becoming optimal when an additional preference is added

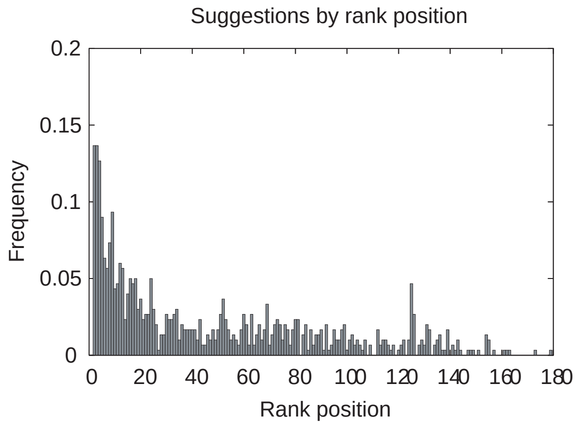

Figure 2: The position of suggestions in the overall ranking.

(*lookahead strategy*). We call them *model-based* suggestion strategies because they specifically choose examples to stimulate the expression of additional preferences based on the current preference *model*. We evaluated our approach with user studies, showing that significant gain in accuracy can be achieved by the use of the suggestion strategies.

Pareto-optimality is chosen to limit the effect of the numerical error in the internal representations of prerences. An option *dominates* another one if it is not worse according to all preferences and strictly better for at least one preference; an option is **Pareto-optimal** if it is not dominated by any other.

In [9], we discussed the motivation behind our intuition and presented an analysis of real user interaction logs.

2. APPROXIMATIONS

The method requires to compare each option with its dominators to check if the dominance can be broken if a new preference is stated on any of the other attributes. The overall complexity is $O(n^2)$, where n is the number of available options. For large databases, we propose the following approximations:

- to select suggestions from the top k_b options
- to replace Pareto-dominance with utility dominance
- to assume dominating options as a fixed number of options at the top

The first approximation arises from the observation that in most of the cases suggestions are options that are highly ranked (Figure 2). The second approximation considers the total ordering established by the combination function defined in the preference modeling formalism, such as a weighted sum; this method has still quadratic complexity, even if faster in practice. The third approximation tests each option against a fixed number k_d of best options to check if their dominance could be broken; this method is much faster with a complexity linear in the database size.

2.1 Evaluation of the Approximations

We evaluated the approximated techniques on the ability to find Pareto-optimal options considering the unknown preferences. We ran simulations with the apartment database and with randomly generated data. We generated random models consisting of 2 to 7 preferences and we calculated the number of times the method successfully fulfilled our lookahead strategy (the *hit rate*) when one preference is not stated yet (i.e. the frequency of finding, among the suggestions selected, an option that became Pareto-optimal by the addition of one among the missing preferences).

method	note	hit rate
model-based suggestions		77%
approximation 1	$k_b = 50\%$	74%
	$k_b = 25\%$	72%
	$k_b = 12.5\%$	64%
approximation 2	utilitarian	66%
approximation 3	k_d=3	23%
	k_d=6	31%
	k_d=12	49%
random suggestions		12%
no suggestions	more candidates	31%

Table 1: The hit-rate according to our look-ahead principle for the different approximation strategies and the complete method of generation of suggestions. For comparison, we show the case in which we simply display more candidates.

3. FINAL REMARKS

While user studies show that tools based on examples are effective (decision accuracy up to 80%, [6]), only few web sites currently support preference based search. The main issue is that such personalized search tools are hard to implement for large databases and user populations.

We presented a framework for preference-based search that provide suggestions to the user to stimulate preference expression and we presented different approximations to make our approach scale to large databases. These are discussed in more detail in [8].

4. REFERENCES

[1] R. Fagin. Fuzzy queries in multimedia database systems. In *PODS '98*, pages 1–10, USA, 1998. ACM Press.

[2] G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: The automated travel assistant. In *User Modeling '97*, 1997.

[3] J. Payne, J. Bettman, and E. Johnson. *The Adaptive Decision Maker*. Cambridge University Press, 1993.

[4] P. Pu and B. Faltings. Enriching buyers' experiences: the smartclient approach. In *SIGCHI conference on Human factors in computing systems*, pages 289–296. ACM Press New York, NY, USA, 2000.

[5] P. Pu and P. Kumar. Evaluating example-based search tools. In *ACM Conference on Electronic Commerce (EC'04)*, 2004.

[6] P. Pu, P. Viappiani, and B. Faltings. Increasing user decision accuracy using suggestions. In *CHI 2006*, April 2006.

[7] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Dynamic critiquing. In P. Funk and P. A. González-Calero, editors, *ECCBR*, volume 3155 of *Lecture Notes in Computer Science*, pages 763–777. Springer, 2004.

[8] P. Viappiani and B. Faltings. Design and implementation of Preference-based Search. Technical report. LIA-2006-04, EPFL, Lausanne, Switzerland, 2006.

[9] P. Viappiani, B. Faltings, and P. Pu. The lookahead principle for preference elicitation: Experimental results. In *Seventh International Conference on Flexible Query Answering Systems (FQAS)*, 2006.