Evaluating Preference-based Search Tools: a Tale of Two Approaches

Paolo Viappiani[1], **Boi Faltings**[1] **and Pearl Pu**[2]
1) Artificial Intelligence Laboratory (LIA)
2) Human Computer Interaction Group(HCI)
Ecole Polytechnique Fdrale de Lausanne (EPFL)
1015 Lausanne, Switzerland
paolo.viappiani@epfl.ch

Abstract

People frequently use the world-wide web to find their most preferred item among a large range of options. We call this task *preference-based search*. The most common tool for preference-based search on the WWW today obtains users' preferences by asking them to fill in a form. It then returns a list of items that most closely match these preferences. Recently, several researchers have proposed tools for preference-based search that elicit preferences from the critiques a user actively makes on examples shown to them. We carried out a user study in order to compare the performance of traditional preference-based search tools using form-filling with two different versions of an example-critiquing tool. The results show that example critiquing achieves almost three times the decision accuracy, while requiring only slightly higher interaction effort.

Introduction

People often use the world wide web to search for their most preferred item, such as a computer, a camera, an apartment, or a flight. However, keyword search is often too imprecise, returning hundreds or thousands of results that users cannot practically process themselves. Instead, it is more effective to guide the search with user preferences, expressed as preferred values on the attributes of an item. Then a database system can find the ideal item based on these preferences. We call this task *preference-based search*.

For structured items, the most common way to obtain the user's preferences is to ask the user to fill in a form or to answer a set of elicitation questions. Such a process is used, for example, when searching for flights on the most popular travel web sites,[1] [2] or when searching for apartments, used cars or health insurance using the Swiss site Comparis[3]. The comparis site (see Figure 1) presents the users with a form on which they fill in their preferences, and then shows (in possibly many pages) all results that satisfy these preferences. The user can go back to the form to modify the preferences and obtain different search results. We call this preference elicitation approach the *form-filling* approach. It has been used in essentially unchanged form for years, despite studies such as (Equity 2000) which showed that only 18% of users feel that they find the best solution using such a tool.

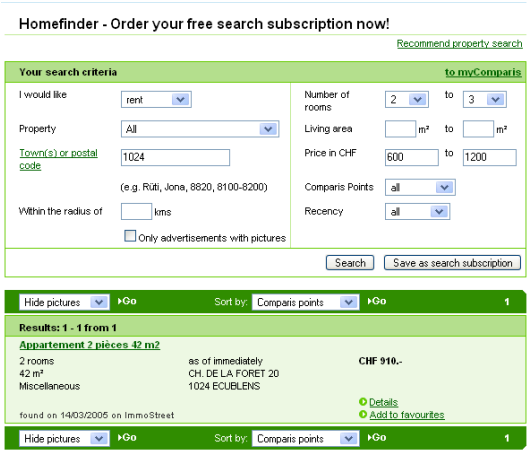

Figure 1: Example of preference-based search using the form-filling approach.

When the user starts a search, she typically has only few well-defined preferences. According to behavioral studies (Payne, Bettman, & Johnson 1993; Slovic 1995; Tversky 1996), many of the preferences are *constructed* when considering specific examples. Thus, with a form-filling approach, the preference model may be quite complete, but does not guide the system to find the most accurate results. Similar effects were also documented for collaborative filtering, where allowing users to rate items of their own choice led to more accurate results than making them rate items selected by the system to optimize coverage (McNee *et al.* 2003).

It is thus reasonable to expect more accurate search results if preferences are expressed on users' own initiative. Moreover, people often express preferences as reactions on the example solutions shown to them, such as "I like this digital camera, but find something less expensive." Therefore, researchers have proposed to elicit user preferences as *critiques* on examples, and use this feedback to direct the next search cycle. This approach, termed example-critiquing, is

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

[1] http://www.travelocity.com/
[2] http://www.expedia.com
[3] http://www.comparis.ch/

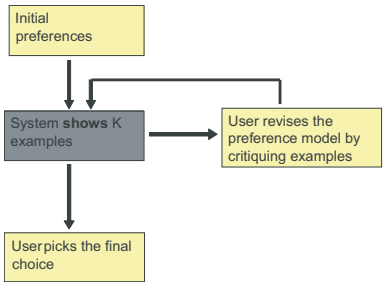

Figure 2: *The example-critiquing interaction model. The dark box is the computer's action and the other boxes show actions of the user.*

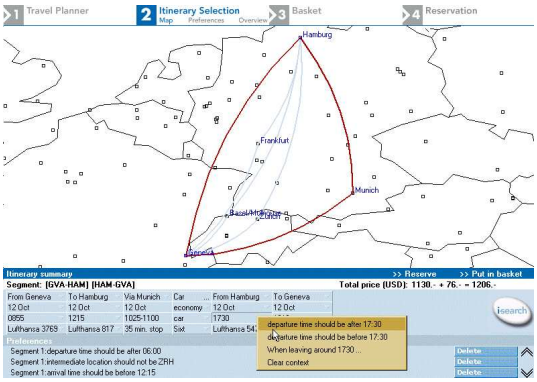

Figure 3: *Isy-travel is an example-critiquing tool for planning business trips. Here the user is planning a one-day trip from Geneva to Hamburg. The preference model is shown at the bottom, and the user can make critiques by clicking on features of the example currently shown.*

illustrated in Figure 2. A difficulty is that since the initiative to express critiques lies with the users, they must be motivated to express them. A major contribution of this paper is a method for generating *suggestions* that give the user an idea of the opportunities that other options may present. Through a user study, we confirm the advantages of example-critiquing over form-filling, and the importance of suggestions for example-critiquing. Results show that decision accuracy can be increased from 25% for form-filling to 70% for example-critiquing.

Related Work

To our knowledge, the earliest work on example-critiquing has been reported by (Tou *et al.* 1982), and has been picked up again in the late 1990s by several researchers. Figure 3 shows an example taken from Isy-Travel, a commercial tool for business travelers (Pu, Faltings, & Torrens 2004). Here, the user is shown options that best match the current preference model. The idea is that an example is either the most preferred one, or there are some aspects a user wants to improve. 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. This form of example-critiquing has been proposed by various researchers, starting with the ATA system of Linden et al. (Linden, Hanks, & Lesh 1997) and SmartClient (Pu & Faltings 2000), and more recently with incremental critiquing (Kevin McCarthy 2005).

In another form of example-critiquing, search proceeds as *navigation* from one example to another, where the current example is considered to be the most preferred at that point. Critiques of the current example then lead to another example that improves on the aspect that the user has critiques while being as similar as possible to the current example. Performance can be improved by providing users with compound critiques (Reilly *et al.* 2004). This form of example-critiquing was first proposed in the FindMe systems (Burke, Hammond, & Young 1997; Burke 2002), and more recently also used in the ExpertClerk system (Shimazu 2001) and in dynamic critiquing (McCarthy *et al.* 2005).

Recently, researchers have considered whether example-critiquing should present the user only with examples that match the current preferences best, or with a diversity of options that show the available choices and thus stimulate the user to express more preferences. Such *suggestions* were first considered by Linden et al. (Linden, Hanks, & Lesh 1997), and subsequently by other researchers who were studying the potential need for diversity, for example (Shimazu 2001; McSherry 2002; Smyth & McGinty 2003; McGinty & Smyth 2003). Suggestions can be used with any interface design that allows incremental revision of preferences.

We now describe the techniques for example-critiquing and suggestion generation in more detail, and show a tool called `FlatFinder` that implements the various example-critiquing techniques on an apartment database. We used this tool to compare the performance of the different methods with real users.

Implementing example-critiquing

In this paper, we consider example-critiquing with an explicit preference model. It is an iterative process of showing examples, eliciting critiques and refining the preference model. Thus, the essential issues concern preference modeling and the generation of examples to display. We address both issues in turn, focusing on strategies for generating suggestions.

Modeling Items and Preferences

We assume that items are modelled by a fixed set of n *attributes* $A = \{A_1, .., A_n\}$ that each take values in associated domains $D_1, .., D_n$. Domains can be *qualitative*, consisting of an enumerated set of possibilities, or *numeric*. Each option o is characterized by the values $a_1(o), ..., a_n(o)$ of the attributes.

The user's *preferences* are assumed to be defined on individual attributes[4] and independent of one another:

[4]For preferences over a combination of attributes, such as the total travel time in a journey, we assume that the model includes additional attributes that model these combinations.

Definition 1 *A* preference r *is an order relation* $\preceq_r$ *of the values of an attribute* a*;* $\sim_r$ *expresses that two values are equally preferred. A* preference model *R is a set of preferences* $\{r_1, .., r_m\}$.

As a preference r always applies to the same attribute a_z, we simplify the notation and apply $\preceq_r$ and $\sim_r$ to the options directly: $o_1 \prec_r o_2$ iff $a_z(o_1) \prec_r a_z(o_2)$. We use $\prec_r$ to indicate that $\preceq_r$ holds but not $\sim_r$.

For a practical preference-based search tool, it is impractical to model the order relations as such. Instead, it is more convenient to express them by a numerical *penalty* (or cost) function:

Definition 2 *A penalty function* c*,* $d_k \rightarrow \Re^+$*, maps from an attribute* a_k *to a number that gives a penalty(cost) of that attribute value to the user such that whenever* $o_1 \succ_{r_i} o_2$ *(*o_1 *is preferred to* o_2*),* $c_i(o_1) < c_i(o_2)$.

We assume that the penalty functions correctly express the user's *ceteris paribus* preferences, i.e. that for any pair of options o_1 and o_2 that are identical in all preferences except preference r_i, the user prefers o_1 over o_2 if and only if $c_i(s_1) < c_i(s_2)$.

An overall ranking of options can be obtained by combining the penalty functions for all stated preferences. In our systems, we combine them using a weighted sum, which corresponds well with standard multi-attribute utility theory (Keeney & Raiffa 1976). Thus, if $\mathcal{R}_c$ is the set of the penalty functions, we compute the cost $C(o) = \sum_{c_i \in \mathcal{R}_c} w_i \cdot c_i(o)$. Option o_1 is preferred over option o_2 whenever it has a lower cost, i.e. $C(o_1) < C(o_2)$.

Note that a user will not state preferences in a numerically precise function, but only qualitatively. We map these qualitative statements into parameterized functions that are standardized to fit average users. These are chosen with respect to the application domain.

For example, in travel planning there are attributes relating to time. Thus, the system would provide functions to characterize "later than x" or "earlier than x" by graded step functions that penalize times that significantly violate the preference. Other functions would allow to express airline preference by a function that gives a penalty to all but a certain value of an airline attribute. More details about preference modeling for example-critiquing are discussed in (Pu & Faltings 2004).

Generating Optimal Examples

Preference-based search looks for the option that best satisfies the preference model among some well-defined set of choices. This can be a fixed database of options, such as in the apartment search example we will discuss later, or it can be constructed through a configuration system, as in a travel planning application.

In the first case, the best options can be found by sorting the database items according to their cost. This is known as the top-k query in the database community (Fagin 1998). In the second case, optimization algorithms such as branch-and-bound can be used to find the best options during the configuration process.

Note that as users' qualitative preferences are mapped into standardized functions, the resulting cost function is inaccurate and thus results in an inaccurate ordering. A common approach in many web search applications is to compensate for this inaccuracy by showing not just one, but a set of k options. Faltings et al. in (Faltings, Torrens, & Pu 2004) show that given a bound on the difference between the standardized and a user's actual penalty function, it is possible to compute a minimum k such that the user can find the truly best option among this set of k possibilities. Interestingly, while k grows with the inaccuracy of the functions and the number of preferences, it is independent of the total number of options available, so that the approach scales even for search in very large collections.

The Need for Suggestions

In an example-critiquing interaction, user's preferences are *volunteered*, not elicited. This means that users are never forced to answer questions about preferences they do not yet possess.

We can expect users to state additional preference as long as they perceive them to lead to a better solution. The process ends when users can no longer see potential improvements by stating additional preferences and have thus reached an optimum. However, this could be a local optimum. For example, a user looking for a notebook computer might start looking for a low price. Assume that all cheap models weigh about 3 kg. If only those are presented, they will all have about the same weight, and it may not occur to her to look for lighter models. The influence of the current examples prevents the user from refocussing the search in another direction. It is called the *anchoring effect* (Tversky 1974) in behavioral decision theory.

Several authors have proposed to improve the examples shown to users by suggesting items that can potentially bring the user out of such a local optimum (Linden, Hanks, & Lesh 1997; Smyth & McGinty 2003; Shimazu 2001; McSherry 2002; McGinty & Smyth 2003). They have used various heuristic strategies such as showing examples with extreme values in certain attributes, or generally ensuring a certain diversity.

However, the motivation for suggestions is to make a user state additional preferences. We can assume that the user is minimizing her own effort and will add preferences to the model only when she expects them to have an impact on the solutions. This is the case when:

- the user can see several options that differ in a possible preference, and
- these options are relevant, i.e. they are reasonable choices, and
- they are not already optimal.

Otherwise, stating an additional preference is irrelevant: when all options would evaluate the same way or when the preference only has an effect on options that would not be eligible anyway, stating it would be wasted effort. This has led us to the following principle, which we call the *look-ahead* principle, as a basis for our model-based suggestion strategy:

Suggestions should not be optimal under the current preference model, but should provide a high likelihood of optimality when an additional preference is added.

This is a heuristic principle based on assumptions about human behavior that we cannot prove. However, using suggestions is justified by the fact that it works very well in user studies, as we report later in this paper.

Model-Based Suggestion Strategy

Our strategy of choosing suggestions is *model-based* because the current preference model guides the selection of examples to stimulate the expression of additional preferences.

Recall that preferences are modeled by standardized functions that correctly reflect the preference order of individual attribute values but may be numerically inaccurate. When looking for the best solution, we can compensate for this inaccuracy by showing a set of possibilities. However, when generating suggestions we would like to use a model that is not sensitive to this numerical error. We thus use the qualitative combination concepts of *dominance* and *Pareto-optimality*:

Definition 3 *An option o is* dominated by *an option o' with respect to R if and only if for all $r_i \in R$, $o \preceq_{r_i} o'$ and at least one $r_j \in R$, $o \prec_{r_j} o'$. We write $o \prec_R o'$. We can also say that o is* dominated*, without specifying o'.*

Note that we use the same symbol $\prec$ for both individual preferences and sets of preferences.

Definition 4 *An option o is* Pareto-optimal *(PO) if and only if it is not dominated by any other option.*

Pareto-optimality is the strongest concept that would be applicable regardless of the numerical details of the penalty functions.

In our applications, users initially state only a subset R of their true preference model $\overline{R}$. When a preference is added, dominated options with respect to R can become Pareto-optimal. The following observation is the basis for evaluating the likelihood that a dominated option will become Pareto-optimal:

Proposition 1 *A dominated option o' with respect to R becomes Pareto-optimal with respect to $R \cup r_i$ (a new preference r_i is added), if and only if o' is strictly better with respect to r_i than all options that currently dominate it: $o' \succ_{r_i} o, \forall\, o \in O_R^+(o')$.*

where we use the concept of *dominating set*: given one option o, the set of all options that dominate it: $O_R^+(o) = \{o' \in O : o' \succ_R o\}$. We write $O^+(o)$ if it is clear from the context which is the set R of preferences we are considering.

Proposition 1 provides a heuristic for implementing the look-ahead principle: choose suggestions that have the highest likelihood of breaking all dominance relations with options in its dominating set when a new preference is added.

Letting P_{a_i} be the probability that a user has a preference on attribute a_i, we consider the probability p_d that it makes option o escape dominance by another option o_+:

$$p_d(o, o_+) = 1 - \prod_{a_i \in A_u} (1 - P_{a_i} \delta_i(o, o_+))$$

where $o_+ \in O^+(o)$, the dominating set of o, and δ_i is a heuristic estimation of the probability that o is better than o^+ according to a hidden preference on attribute a_i, which could breaking the dominance relation.

For attributes with numeric values, we use a normalized difference for interval domains: the chances that a new preference will treat o_1 and o_2 differently is directly proportional to the difference between their values. For qualitative attributes, it is sufficient to detect if attributes are different: in this case there are equal chances that o_1 will be preferred over o_2 and vice versa so that $\delta = 0.5$. If the values are the same, the dominance relation cannot be broken by a preference on this attribute and so $\delta = 0$.

For avoiding dominance, we need to determine the probability than an option breaks the domination of *all* dominating options (i.e. become Pareto-optimal), not just one of them. Since a new preference is always stated on some attribute, this means that there has to be one attribute where all the dominating options have different values than the suggestion. For interval domains, we assume that user preferences are monotone, so that the requirement is that the attribute value in the suggestion has be lower or higher than in all options of the dominating set. We use $\delta_i(o, O^+)$ to denote the analogous distance to δ_i, but to the closest value of any dominating option.

The probability of breaking all the dominance relations simultaneously by stating one new preference is then:

$$p_d(o, O+) = 1 - \prod_{a_i \in A_u} (1 - P_{a_i} \delta_i(o, O^+)) \qquad (1)$$

If we assume that the user has only one hidden preference, we can use the following simplification:

$$p_d(o, O+) = \sum_{a_i \in A_u} P_{a_i} \delta_i(o, O^+) \qquad (2)$$

which is also a good approximation when the probabilities for additional preferences are small.

Following the lookahead principle, the options for which $p_d(o, O+)$ is greatest are chosen as the best suggestions.

Example

As an example to illustrate the various approaches, consider selecting a flight among the following set of options:

	fare	arrival	airport	airline
o_1	250	14:00	INT	B
o_2	300	9:00	INT	A
o_3	350	17:30	CITY	B
o_4	400	12:30	CITY	B
o_5	550	18:30	CITY	B
o_6	600	8:30	CITY	A

For simplicity, assume that options are modelled by just 4 attributes: fare, arrival time, departure airport, and the airline.

Initially, the user starts with a preference for the lowest price. Assume that the user also has two other, hidden preferences:

- arrive by 12:00;

- leave from the CITY airport, which is much closer than the INTernational airport.

The best choice for this user will be o_4 which gives a reasonable tradeoff among the objectives.

In a form-filling or wizzard approach, the tool might also ask the user for a preference on the airline. Even though she does not have any preference on the airline, she might believe that airline A operates more flights from the CITY airport and thus formulate a *means objective* for airline A. When this preference is added to the others, it now makes options o_2 or even o_6 the most attractive, thus keeping the user from finding her best choice. We believe that this effect is responsible for the poor performance of form-filling in the user studies we report later in this paper.

Now consider an example-critiquing approach. Given the initial preference on price, the tool will start by showing only option o_1. In a tool based on navigation without an explicit preference model, the tool uses the current best example as the starting point and finds the one that is closest to it while also satisfying the user's critique. In this case, the user might first critique the fact that the arrival time is too late, leading to o_2, and then the fact that the departure is from the INTernational airport, leading to o_6 as the most similar option leaving from the CITY airport. How, the user might critique the fact that this option is too expensive, and get back to o_2. In this cycle, the best option, o_4, will never be discovered, since it requires a tradeoff among the different preferences. The importance of such tradeoffs has been pointed out for example in (Pu & Chen 2005).

In a tool with an explicit preference model, o_4 will be determined as the best option once both hidden preferences have been added. However, the user might need to be motivated to ask for these options by seeing suggestions that point them out as opportunities.

Consider first the strategy of proposing options with extreme attribute values proposed by Linden et al. (Linden, Hanks, & Lesh 1997). For the departure time, o_5 is the earliest and o_6 the latest departure. Are these good suggestions to make the user understand the opportunities offered by the available options? Consider:

- unless the arrival has to be after 17:30, o_3 is a much cheaper, thus more attractive option than o_5.
- unless the arrival has to be before 9:00, o_2 is much more attractive than o_6.

Showing these as suggestions would give the user an impression that a different arrival time implies significant compromises on her preference for low cost, which in reality is not true. Thus, she might never state her preferences on time given these suggestions.

The strategy of showing maximally diverse options gives the same result: as o_1 is the best option, the options that are most different are again o_5 and o_6. We have already seen that these are not the best to motivate preferences on arrival time, but they are not motivating for other attributes either:

- if the user prefers the CITY over the INTernational airport, o_3 is a much cheaper option that already allows this.
- if the user prefers airline A, o_2 allows this at a much lower cost.

The model-based suggestion strategy takes this observation into account by evaluating options with respect to others that might dominate them. The following table shows the normalized differences δ_i to the attribute range of the dominating options as well as the probability p of breaking dominance with all of them, assuming that the user has a probability P_{a_i} of 0.5 of having a preference on each of the three attributes that do not yet have a preference.

	fare (a_1)	$arrival$ (a_2)	δ_2	airport (a_3)	δ_3	airline (a_4)	δ_4	p
o_1	250	14:00	-	INT	-	B -	-	
o_2	300	9:00	0.5	INT	0	A	0.5	0.437
o_3	350	17:30	0.35	CITY	0.5	B	0	0.381
o_4	400	12:30	0	CITY	0	B	0	0
o_5	550	18:30	0.1	CITY	0	B	0	0.05
o_6	600	8:30	0.05	CITY	0	A	0	0.025

We can see that now, options o_2 and o_3 are considered the best suggestions. This is due to the fact that they are quite competitive given the known preference on price, and show the availability of other attribute values without unduly compromising the already known objectives. Thus, they are most likely to stimulate the additional preference expression that we require.

Note that the best option - o_4 - is not among the suggestions. However, it will become the highest-ranked option once the hidden preferences have been stated. This illustrates that the function of suggestions is to motivate additional preferences, but not necessarily to provide the best choices by themselves.

Evaluation with User Studies

We carried out a user study to validate the hypothesis that obtaining preferences incrementally through example-critiquing increases decision accuracy over the form-filling approach. We also wanted to confirm that the presence of suggestions was important. The study used FlatFinder, a tool for finding student accommodation using example-critiquing.

We performed two experiments. The main experiment was carried out among 3 different groups of users, each using a different tool for their search with no previous exposure to the options. A second experiment was a within-subject experiment where the same subjects who had used the form-filling tool also used the example-critiquing tool. This was mainly used to confirm that the first evaluation result did not carry any user bias.

Between-Group Experiment

The between-group experiment used 20 users in each of the three groups. They were students looking for new or better housing and thus were very motivated to carry out the experiment. Each group used one of the three versions of the FlatFinder system:

1. a form-filling interface similar to that used in existing web sites such as `http://www.comparis.ch/`. The form contains all preferences that can be used when

Version	Accuracy	Average Time
Form-filling	0.25	2:45
Iterated form-filling	0.35	5:30
example-critiquing	0.45	8:09
example-critiquing (with suggestions)	0.70	7:39

Table 1: Accuracy and task execution time for different versions (between-groups experiment).

searching for housing, and the tool returns 6 options that best fit the preferences;

2. the example-critiquing interface, where preferences are stated on the user's initiative. The tool returns 6 best options according to the current preferences;
3. the same example-critiquing interface, but returning 3 best options and 3 best suggestions according to the probabilistic suggestion strategy.

The subjects first selected the most preferred options using their respective version of the FlatFinder tool. For the first group, we recorded both the best option selected in the first use, and after iterated use of the form-filling. At the end of the experiment, we asked the subjects to carefully go through the entire list of available options (about 150) and select their truly most preferred choice, a process that took them on average about 30 minutes. We can thus measure the decision accuracy as the fraction of times that the choice found using the tool agrees with that found during this detailed examination. This measure, also called the *switching rate*, is the most widely accepted measure of quality for product search tools in the marketing literature (see for example (Haubl & Trifts 2000)).

The results of the experiment are shown in Table 1. They clearly show the variation in performance for the different versions of the tool. Using the traditional form-filling approach, only 25% of users found their most preferred solution. This fraction only increased to 35% when they could repeat the use of this interface as many times as they liked. On the other hand, example-critiquing reached a 45% accuracy. However, the improvement is not statistically significant: the student test gives values of p=0.097 for the difference between form-filling and simple example-critiquing, and p=0.265 for repeated form-filling and simple example-critiquing.

We obtained the strongest increase in accuracy, to 70%, when using example-critiquing with suggestions. This difference is strongly statistically significant with p=0.00176 for single form-filling and p=0.0133 for repeated form filling. This shows that suggestions are critical in improving decision accuracy for example-critiquing tools.

However, as Table 1 also shows, the increased accuracy comes at the expense of a longer interaction time. Partly this is due to the fact that users were not familiar with the example-critiquing interface and thus took some time to get used to it. Interestingly, example-critiquing with suggestions required less interaction time than without suggestion but achieved much higher accuracy, which is another indication of the importance of suggestions.

Version	Accuracy	Average time
Form-filling	0.25	2:45
Form with revision	0.35	5:30
example-critiquing (with suggestions)	0.65	6:02

Table 2: The accuracy achieved by users of the different versions (within-subject experiment).

We attribute the poor performance of form-filling to the fact that people were driven to state many preferences before having considered any of the available options. Subjects that used the form-filling approach started with an average of 7.5 preferences, and only 5 of 20 ever removed any of them. In contrast, subjects that used example-critiquing with suggestions started with an average of only 2.7 preferences, but added an average of 2.6 to reach 5.3 preferences at the end of the interaction. Here, half of the preferences were constructed while considering examples, and the results suggest that they were much more accurate.

(Pu & Chen 2005) reported the significant influence of value tradeoffs, reflected as preference revisions, on decision accuracy. Here, we observe similar correlation: people who found their target item made 6.93 preference revisions on average, whereas those who did not find their item made an average of only 4.51 revisions. The difference is statistically significant with p=0.0439. This confirms a similar correlation, even though our tool did not provide specific support for tradeoffs and revisions as was used in (Pu & Chen 2005).

Within-Subject Experiment

To ensure that the differences in accuracy were not due to the different backgrounds and preference fluency of the users interacting with the tool, we also carried out a within-subject experiment where the same group of 20 users first used the form-filling version, the iterated version, and finally the example critiquing version with suggestions. Note that to avoid overloading the subjects, we did not test example-critiquing without suggestions.

Table 2 shows the resulting accuracy and interaction time. We attribute the slightly lower accuracy and interaction time for example-critiquing to the fact that users were tired during a second interaction, and thus less likely to perform a deep interaction with the tool. The results however clearly show that example-critiquing gives a very significant improvement in decision accuracy. The student test has a significance of p=0.00506 that example-critiquing with suggestions provides more accuracy than single form-filling, and p=0.03 that it provides more accuracy than repeated form-filling.

Conclusion

Web users are commonly faced with the task of searching for their most preferred item using a search tool. Most tools on the WWW require users to express preferences by filling forms, and then provide the best matching solutions. While

this approach works well for decision makers who know their preferences before the interaction, it does not support the process of preference construction for users who are not familiar with the choices.

We have shown the incremental approach of preference construction via *example-critiquing* and demonstrated through user studies that it provides much higher decision accuracy than the form-filling approach. It appears that when users are asked to state their preferences without considering the options, many of these preferences are simply inaccurate. Since users are reluctant to retract them, this leads to inaccurate search results. On the other hand, significant gain in accuracy can be achieved by the use of active suggestion strategies to stimulate preference expression and increase its fluency.

With this group of participants, we have also confirmed the results of Pu and Chen (Pu & Chen 2005) that a higher decision accuracy is achieved by users who perform more preference revisions. We are considering ways to combine active tradeoff support with the incremental preference elicitation we use here to see if even higher accuracy can be obtained.

For operators of e-commerce sites, more accurate search tools should help increase consumer satisfaction, attract more repeat visitors, and increase conversion rate and revenue.

Acknowledgments

We thank Vincent Schickel-Zuber for his contribution in the implementation of the web version of FlatFinder. This work has been funded by the Swiss National Science Foundation under contract No. 200020-103421.

References

Burke, R. D.; Hammond, K. J.; and Young, B. C. 1997. The FindMe approach to assisted browsing. *IEEE Expert* 12(4):32–40.

Burke, R. 2002. Interactive critiquing for catalog navigation in e-commerce. *Artificial Intelligence Review* 18(3-4):245–267.

Equity, M. S. D. W. 2000. Transportation e-commerce and the task of fulfilment. *Research Europe*.

Fagin, R. 1998. Fuzzy queries in multimedia database systems. In *PODS '98: Principles of database systems*, 1–10. New York, NY, USA: ACM Press.

Faltings, B.; Torrens, M.; and Pu, P. 2004. Solution generation with qualitative models of preferences. In *Computational Intelligence*, 246–263(18). ACM.

Haubl, G., and Trifts, V. 2000. Consumer decision making in online shopping environments: The effects of interactive decision aids. *Marketing Science* 19(1):4–21.

Keeney, R. L., and Raiffa, H. 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* John Wiley and Sons, New York.

Kevin McCarthy, Lorraine McGinty, B. S. J. R. 2005. A live-user evaluation of incremental dynamic critiquing. In *6th International Conference on Case-Based Reasoning*, volume 3620, 339–352. Springer LNAI.

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

McCarthy, K.; McGinty, L.; Smyth, B.; and Reilly, J. 2005. On the evaluation of dynamic critiquing: A large-scale user study. In Veloso, M. M., and Kambhampati, S., eds., *AAAI*, 535–540. AAAI Press AAAI Press / The MIT Press.

McGinty, L., and Smyth, B. 2003. On the role of diversity in conversational recommender system. In *ICCBR 2003*, 276–290. LNAI 2689.

McNee, S. M.; Lam, S. K.; Konstan, J. A.; and Riedl, J. 2003. Interfaces for eliciting new user preferences in recommender systems. In *User Modeling 2003*, LNCS 2702, 178–187. Springer.

McSherry, D. 2002. Diversity-conscious retrieval. In Craw, S., and Preece, A. D., eds., *ECCBR*, volume 2416 of *Lecture Notes in Computer Science*, 219–233. Springer.

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

Pu, P., and Chen, L. 2005. Integrating tradeoff support in product search tools for e-commerce sites. In Riedl, J.; Kearns, M. J.; and Reiter, M. K., eds., *ACM Conference on Electronic Commerce*, 269–278. ACM.

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

Pu, P., and Faltings, B. 2004. Decision tradeoff using example-critiquing and constraint programming. *Constraints: An International Journal* 9(4).

Pu, P.; Faltings, B.; and Torrens, M. 2004. Effective interaction principles for online product search environments. In *Proceedings of the 3rd ACM/IEEE International Conference on Web Intelligence*. IEEE Press.

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

Shimazu, H. 2001. Expertclerk: Navigating shoppers buying process with the combination of asking and proposing. In *(IJCAI'01)*, volume 2, 1443–1448.

Slovic, P. 1995. The construction of preference. *American Psycologist* 50.

Smyth, B., and McGinty, L. 2003. The power of suggestion. In *IJCAI-03 Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003*, 127–132.

Tou, F. N.; Williams, M. D.; Fikes, R.; Jr., D. A. H.; and Malone, T. W. 1982. Rabbit: An intelligent database assistant. In *AAAI*, 314–318.

Tversky, A. 1974. Judgement under uncertainity: Heuristics and biases.

Tversky, A. 1996. Contrasting rational and psychological principles in choice. *Wise Choices: Decisions, Games and Negotiations*.