Increasing User Decision Accuracy using Suggestions

Pearl Pu
Human Computer Interaction Group(HCI)
Ecole Polytechnique Fédérale de Lausanne (EPFL) Station 14
1015 Lausanne, Switzerland
pearl.pu@epfl.ch

Paolo Viappiani
Artificial Intelligence Laboratory (LIA)
Ecole Polytechnique Fédérale de Lausanne (EPFL) Station 14
1015 Lausanne, Switzerland
paolo.viappiani@epfl.ch

Boi Faltings
Artificial Intelligence Laboratory (LIA)
Ecole Polytechnique Fédérale de Lausanne (EPFL) Station 14
1015 Lausanne, Switzerland
boi.faltings@epfl.ch

ABSTRACT

The internet presents people with an increasingly bewildering variety of choices. Online consumers have to rely on computerized search tools to find the most preferred option in a reasonable amount of time. Recommender systems address this problem by searching for options based on a model of the user's preferences.

We consider example critiquing as a methodology for mixed-initiative recommender systems. In this technique, users volunteer their preferences as critiques on examples. It is thus important to stimulate their preference expression by selecting the proper examples, called suggestions. We describe the look-ahead principle for suggestions and describe several suggestion strategies based on it. We compare them in simulations and, for the first time, report a set of user studies which prove their effectiveness in increasing users' decision accuracy by up to 75%.

Author Keywords

Recommender systems, consumer decision support, example critiquing interfaces, user evaluation of interfaces.

ACM Classification Keywords

H.1.2 [Models and Principles]: User/Machine Systems - human factors, software psychology;

H.5.2 [Information Interfaces and Presentation]: User Interfaces - evaluation/ methodology, graphical user interfaces.

INTRODUCTION

People increasingly face the difficult task of having to select the best option from a large set of multi-attribute alternatives, such as choosing an apartment to rent, a notebook computer to buy, or financial products in which to invest. Knowledge- and utility-based recommender systems are tools that help people find their most desired item based on a model of their preferences [2–4, 13, 18]. For their performance, it is crucial that this preference model be as accurate as possible. This poses new challenges for human-computer interaction at the cognitive level that have been poorly addressed so far, but are key to the user success rate of such systems on e-commerce sites.

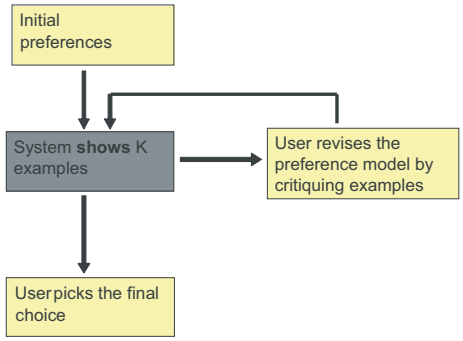

Figure 1. Example critiquing interaction. The dark box is the computer's function, the other boxes show actions of the user.

Utility theory provides a solid mathematical foundation for recommendations [5]. However, it assumes complex preference models that cannot be obtained in e-commerce scenarios because people are not willing to go through lengthy preference elicitation processes. Furthermore, they are usually not very familiar with the available products and their characteristics. Thus, their preferences are not well established, but *constructed* while learning about the available products [11]. To allow such construction to take place, users must be able to explore the space of possible options while building their preference model.

A good way to do this is through a mixed-initiative system based on *example critiquing* (see Figure 1). Example critiquing was first introduced by [25] and works by showing k examples to a user in each interaction cycle. If the target item is not among the k examples, then a set of user critiques will be collected to refine the existing model. Example critiquing allows users to express preferences in any order, on any criteria, and with any effort they are willing to expend [15]. It has been employed by a number of product search and recommender tools [2, 6, 13, 19–21].

CHI 2006, April 22-27, 2006, Montréal, Québec, Canada.

In an example critiquing interaction, user's preferences are *volunteered*, not elicited: users are never forced to answer questions about preferences they might not be sure about. Thus, users will only state preferences that they actually have, and they can tailor the effort they spend on stating their preferences to the importance of the decision that they are making.

RELATED WORK

Example critiquing was first proposed in [25] and has since been used in several recommender systems, such as FindMe [2], ATA [6], SmartClient [13], ExpertClerk [19] and the system of Shearin & Lieberman [20]. The ATA system [6] is particularly important, as it was the first to incorporate the notion of suggestions, which is crucial to our work.

Evaluating example critiquing interfaces has been an active area of research lately. Pu and Kumar [16] showed that example critiquing interfaces enable users to perform decision tradeoff tasks more efficiently with considerably less errors than non-critiquing interfaces. More recently, Pu and Chen [17] showed that the implementation of tradeoff support can increase users' decision accuracy by up to 57%.

In *dynamic critiquing* [18], a popular family of example critiquing interfaces, a metaphor of navigation through the product space is implemented; the interface proposes precomputed critiques (simple and compound) that can be selected by the users. McCarthy et al. [7] showed that users who applied more frequently compound critiques in a critiquing interface were able to reduce interaction cycle from 22 to 6 .

Several researchers [1, 8–10, 21, 22] recognized the need to suggest *diverse* examples in recommender tools. In the context of case-based reasoning [9, 22], algorithms have been proposed to optimize both the similarity to the target (i.e. the optimality) and the diversity of the retrieval set. This approach has been applied to recommender systems, and it has been shown in [8] that such techniques can reduce the length of the recommendation cycle by up to 76%, compared to the pure similarity-based recommender. In [21], diversity is used to implement system recommendations to the query *show me more like this*. Their adaptive search algorithm alternates between a strategy that favors similarity and one that favors diversity (*refocus*).

More recent work on diversity was motivated to compensate for users' preference uncertainty [12], where the utility function is parameterized over a probability distribution, or to cover different topic interests in collaborative filtering recommenders [24].

CONTRIBUTION OF THIS WORK

In our approach, a preference model consists of a user's stated preferred attribute values and their relative importance. When preferences are inferred or constructed from a set of examples, human subjects have been found to favor outcomes based on the superiority of only one or few attributes. This phenomenon is known as the prominence effect [23]. However, a more rational behavior is to evaluate potential candidates based on as many attributes as a user may have using compensatory decision strategies [11]. Therefore, users should be guided to not only express more preferences, but also expand on the number of attributes for which preferred values have been established. The latter, called preference enumeration, is thus an important measure of quality for a preference model. We have found that simply showing examples that are optimal for the current preference model may not be enough to overcome the prominence effect. As the experiments described in this paper show, users are not likely to increase attribute enumeration after interacting with optimal examples.

This observation has led us to extend the example critiquing method to include both

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

In this paper, we take a deeper look at how suggestions should be generated and derive a family of new strategies, called *model-based* strategies. Our final result is an evaluation of the impact of these strategies on decision accuracy through user studies. We define decision accuracy as the likelihood that a user finds the most preferred option when using the tool.

However, to avoid the expense of using each of the different possible suggestion strategies in a study with actual users, we first carried out an evaluation based on simulated users. As the purpose of suggestions is to stimulate expression of preference, we compare different suggestion strategies with respect to user's preference *enumeration*, defined as the number of preferences stated by the user. We also show in the experiments that this preference enumeration is indeed positively correlated with decision accuracy.

The simulations compared 6 different strategies: three from our work, one based on random selection, the strategy proposed by Linden et al. [6], and the diversity strategy as described by McSherry [9]. McSherry's algorithm is a further development based on [22]. The comparison with a random strategy was included to rule out any strategy that would have extraordinarily poor performance, while the strategies of Linden and McSherry represent the strategies that are commonly proposed in the literature. Results suggest that the model-based probabilistic strategy performs best, and it was consequently used in a study with real users.

The study is a within-subject comparative user study where 60 live and 40 recruited users compared two versions of example critiquing systems, one with and one without suggestion interfaces. Results indicate that users were able to state significantly more preferences when using the suggestion interfaces (up to 80% accuracy). More importantly, the user study also indicates that a higher preference enumeration leads to more accurate decisions. Among the 40 re-

cruited users, there is a correlation between the number of preference discovered by the suggestions and the decision accuracy($p = 0.03$).

A DEEPER LOOK AT SUGGESTIONS

The problem faced when using a search tool is that the user has to learn how to state her preferences so that the tool can find her most preferred option. We can assume that she is minimizing her own effort and will add preferences to the model only when she can expect them to have an impact on the solutions. This is the case when:

- she can see several options that differ in a possible preference, and
- these options are relevant, i.e. they could be reasonable choices, and
- these options are not already optimal, so a new preference is required to make them optimal.

In all other cases, stating an additional preference is irrelevant. When all options would lead to the same evaluation, or when the preference only has an effect on options that would not be eligible anyway, stating it would only be wasted effort. This leads us to the following *look-ahead* principle as a basis for suggestion strategies:

> Suggestions should be options that could become optimal when an additional preference is stated.

As a simple example consider searching for a flight between two cities A and B. Options are characterized by the attributes: `<price, arrival time, departure airport>`. For the departure airport, there is a city airport (`CTY`) which is very close to where the user lives and a big international airport (`INT`) which takes several hours to reach. Assume that the user has three preferences in this order of importance:

- the lowest price
- arrive by 12:00
- depart from the city airport

and that she initially only states a preference on the price. The other two preferences remain hidden. Finally, assume that the choice is among the following options:

- f_1: `<200, 13, INT>`
- f_2: `<250, 14, INT>`
- f_3: `<300, 9, INT>`
- f_4: `<600, 8:30, INT>`
- f_5: `<400, 12, CTY>`
- f_6: `<400, 16:30, CTY>`
- f_7: `<900, 18, CTY>`
- f_8: `<280, 15, INT>`

According to the first stated preference (lowest price), the options are ordered $f_1 \succ f_2 \succ f_8 \succ f_3 \succ f_5 = f_6 \succ f_4 \succ f_7$.

Assume that the system shows the 2 most promising ones: f_1 and f_2, the two with lowest price. Here f_1 already dominates f_2 (f_1 is better in all respects) according to the users hidden preferences, so she is unlikely to state any additional preference based on these examples.

A strategy that generates suggestions according to diversity might pick f_7 as suggestion as it is most different from what is currently displayed. However, the user is likely to discard this option, because it is very expensive and arrives very late.

A strategy that chooses examples with extreme values would show one of f_4 or f_7. Neither of them is likely to be taken seriously by the user: f_4 is likely to leave at a very early and inconvenient hour, while f_7 arrives much too late to be useful.

What makes f_7 a bad suggestion to show? From the system point of view, where only the preference about the price is known, f_7 is not a great suggestion because for most of the possible hidden preferences, it is likely to be dominated by f_5 or f_6. If the hidden preference is for the city airport, then f_5 dominates because it is cheaper. If the hidden preference is on arrival time, then only if the user requires an arrival later than 16:30 there is a chance that it will not be dominated by f_6, which is otherwise significantly cheaper.

Without knowing the hidden preferences, good suggestions for this scenario would be f_3, which has a reasonable arrival time without a significantly higher price, f_5 or f_6. These examples differ from f_4 and f_7 in that they have a good chance of becoming optimal for a wide range of possible hidden preferences.

PREFERENCE MODELING

Since the suggestion strategies depend on the preference model that is used in the recommender system, we define the preference model that we assume further in the discussion. We stress that these assumptions are only made for generating suggestions. The preference model used in the recommender system could be more diverse or more specific as required by the application. Also, similar model-based suggestion strategies could be derived for other preference models.

Given a fixed set of n *attributes* $A = \{A_1, .., A_n\}$, an option o is characterized by the values $a_1(o), ..., a_n(o)$ that must belong to the fixed domains $D_1, .., D_n$, which can be explicitly *enumerated* or can be *intervals* of continuous or discrete elements.

The user's *preferences* are supposed to be independent and defined on individual attributes:

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* $\{r_1, .., r_m\}$.

If there can be 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. 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 $\sim_r$.

Depending on the formalism used for modeling preferences, there are different ways of combining the order relations given by the individual preferences r_i in the user's preference model R into a global order of the options. For example, each preference may be expressed by a number and the combination may be formed by summing the numbers corresponding to each preference, or by taking their minimum or maximum.

We can obtain suggestion strategies that are valid with most known preference modeling formalisms by using qualitative optimality criteria based on *dominance* and *Pareto-optimality*:

Definition 2. *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 $o \prec_R o'$ (equivalently we can say that o' dominates o and write $o' \succ_R o$)*

We 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 3. *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 is applicable regardless of the preference modeling formalism used. Our techniques use the concept of *dominating set*:

Definition 4. *The dominating set of an option o is the set of all options that dominate o: $O_R^+(o) = \{o' \in O : o' \succ_R o\}$.*

We will write $O^+(o)$ if it is clear from the context which is the set R of preferences we are considering.

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: $o' \succ_{r_i} o, \forall\, o \in O_R^+(o')$.*

In general, the Pareto-optimal set increases when stating more preferences, as the dominance relation becomes sparser.

MODEL-BASED SUGGESTION STRATEGIES

We propose 3 strategies that we call *model-based* suggestion strategies because they specifically choose examples to stimulate the expression of additional preferences based on the current preference *model*. They use Pareto-optimality to implement the principle stated in the introduction: suggestions should not be optimal yet, but have a high likelihood of becoming optimal when an additional preference is added. An ideal suggestion is an option that is Pareto-optimal with respect to the full preference model $\overline{R}$, but is dominated in R, the partial preference model.

Following Proposition 1, the probability of a dominated option o becoming Pareto-optimal is equal to:

$$p(o) = \prod_{o_+ \in O^+(o)} p_d(o, o_+) \tag{1}$$

where p_d is the probability that a new preference makes o escape the domination relation with a dominating option o_+, i.e. if o is preferred over o_+ according to the new preference. Evaluating this probability exactly requires the probability distribution of the possible preferences, generally not known. Therefore we propose several strategies based on increasingly detailed assumptions about these distributions.

Counting strategy

The simplest strategy, the *counting strategy*, is based on the assumption that p_d is constant for all dominance relations. Thus, we assume:

$$p(o) \;=\; \prod_{o_+ \in O^+(o)} p_d = p_d^{|O^+(o)|}$$

Since $p_d \leq 1$, this probability is the largest for the smallest set $O^+(o)$. Consequently, the best suggestions are those with the lowest value of the following counting metric:

$$F_C(o) = |O^+(o)| \tag{2}$$

Probabilistic strategy

The *probabilistic strategy* finds the best possible estimation of the probability that a particular solution will become Pareto-optimal. p_d (Equation 1) can be written as:

$$\begin{aligned} p_d(o, o_+) &= 1 - \prod_{a_i \in A_u} (1 - P_{a_i} \delta_i(o, o_+) \\ &\approx \sum_{a_i \in A_u} P_{a_i} \delta_i(o, o_+) \end{aligned}$$

where $o_+ \in O^+(o)$, the set of dominators of o, and δ_i is an heuristic estimation of the probability that an hidden prefe-

rence on attribute a_i make o better than o^+ according to that preference, hence escaping the dominance relation.

As a heuristic 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 discrete attributes, it is sufficient to check if the attributes take different values.If so, there will be equal chances that one is preferred over the other and $\delta = 0.5$. If the values are the same, the dominance relation cannot be broken by a preference on this attribute, so $\delta = 0$.

Attribute strategy

The *attribute strategy* considers the fact that for breaking the dominance relation with all options in the dominating set, there has to be one attribute where all dominating options have different values. To express this concept, we define the function *diff*:

Definition 5. *For an attribute a_i and a given option o_1 with dominating set O^+, $diff(o_1, a_i, O^+) = 1$ if:*

- *interval domains: $a_i(o_1)$ should be either greater than or smaller than the attribute values for a_i of all options in O^+*
- *enumerated domains: $a_i(o_1)$ should be different than the attribute values for a_i for all options in O^+*

and 0 *otherwise.*

The reasoning is the following: for interval domains, we assume that preferences are continuous, i.e. the user is likely to prefer values to be larger or smaller than a certain threshold, or as large or as small as possible. This applies to attributes like price or travel time and fits well with the majority of users. For enumerated domains, a new preference may break the dominance relation whenever the attribute has a different value. Then we count the number of attributes for which there are no preferences yet and where all dominating options have a different value:

$$F_A(o) = \sum_{a_i \in A_u} P_{a_i} \text{diff}(a_i, o, O^+(o)) \qquad (3)$$

where A_u is the set of attributes on which no preference has been expressed yet; P_{a_i} is the probability that the user has an unstated preference on attribute a_i. It chooses as suggestions those options with the largest value of this metric.

The example continued

Previously in the example, f_1 and f_2 are shown as candidate optimal examples. We will now consider which options will be chosen by the strategies as suggestions, omitting the calculations.

In the counting strategy, the first suggestion will be f_8 (which is not very interesting because it is very similar to the candidates) followed by f_3 as the second. The attribute strategy selects f_6 as the best suggestion. Its dominators for price (f_1, f_2, f_8, f_3) all depart from a different airport and leave before (external interval), so the *diff* is equal to 1 on both attributes. The attribute strategy cannot choose a second suggestion because all other options have the same values for *diff* on both attributes. The probabilistic strategy chooses f_6 and f_5 since they are both dominated by four options (f_1,f_2,f_8 and f_3) but have high chance of breaking this domination because they significantly differ on the other attributes (they leave from the other airport; f_6 lands few hours after, f_5 before).

Let's assume now that the user has stated her preference about price and time. The candidates will now be f_1 and f_3. The suggestions: the counting strategy will propose f_2 and f_5 (dominated respectively only by f_1 and f_3), the attribute will suggest f_5 (different airport than its dominator, f_3) and the probabilistic will give f_5 and f_6. All suggestion techniques show an example with the city airport, and the user is stimulated to state that as a preference.

SIMULATED USER EXPERIMENTS

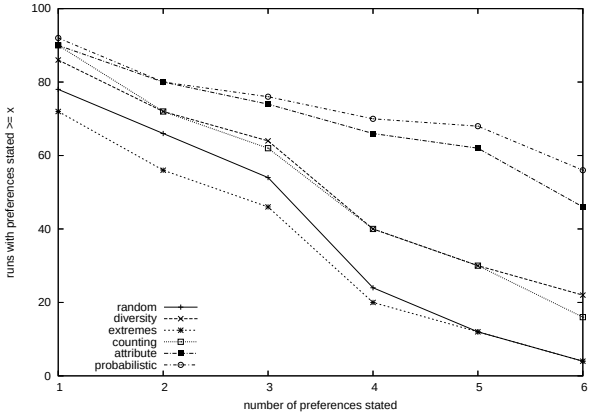

Figure 2. Simulation results on a database of actual apartment offers. For each strategy, we compare the fraction of simulation runs that discover at least x preferences. 100 runs, data-set with 6 attributes and preferences.

The suggestions strategies are heuristics, and it is not clear which of them performs best. Since evaluations with live users can only be implemented with a specific design, we first select the best suggestion strategy by simulating the interaction of a computer generated user with randomly generated preferences. In this way, we can compare the different techniques and select the most promising one for further evaluation. This is followed by real user studies in the next section using the probabilistic suggestion strategy.

In the simulations, users stated a randomly generated set of m preferences on different attributes of available options stored in a database. We are interested in whether the system obtains a complete model of the user's preferences in order to test the objective of the strategies, which is to motivate the user to express as many preferences as possible.

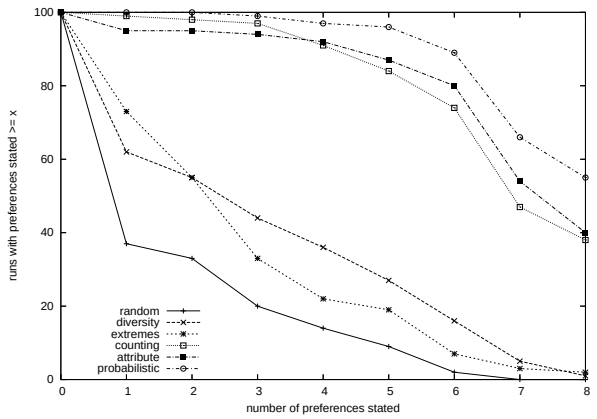

Figure 3. Simulation results for randomly generated problems. For each strategy, we compare the fraction of simulation runs that discover at least x preferences. 100 runs, data-set with 8 attributes and preferences.

The simulated interaction starts with the initial preference (randomly chosen among the m preferences). K options are selected as suggestions according to one of the following strategies: random choice, suggestion of extrema, maximization of diversity (which we include for comparison purposes) and the three model-based suggestions that we propose (counting, attribute and probabilistic). Maximization of diversity consists of selecting the subset of the k most diverse options, so that the diversity (defined as the sum of the difference on all attributes) between each option is maximized [9].

The simulated user behaves according to our model, stating a new preference whenever the suggestions contain an option that would become optimal if such a preference were added to the user model. The interaction continues until either the user model is complete or the simulated user states no further preference. Note that when the complete preference model is discovered the user finds the most wanted option.

The results of the simulation for a catalog of student accommodations (160 options, 10 attributes) are summarized in Figure 2. It shows the percentage of runs (out of 100) that discover at least x out of the 6 preferences in the complete user model. We see that the suggestion strategies provide a marked increase in the number of preferences that are uncovered, and, in particular, the model-based strategies perform best.

In another test, we ran the same simulation for a catalog of 100 randomly generated options with 9 attributes and 9 preferences (one is the initial preference, and 8 are yet to be discovered). The results are shown in Figure 3. We can see that random and extreme strategies now perform very poorly and model-based strategies appear much better. Also, the difference among the three model-based approaches is smaller: the counting strategy performs only slightly worse than the attribute and probabilistic strategies. This occurs because there is no correlation between the attributes.

We investigated the impact of the number of preferences, the number of attributes and the size of the data set. Surprisingly we discovered that the number of attributes only slightly changes the results. Keeping the number of preferences constant at 6, we varied the simulations on the number of attributes set to 6,9,and 12 respectively. The fraction of runs (with 100 total runs) that discovered all the preferences varied for each strategy and simulation scenario by no more than 5%.

We were surprised by the fact that the strategy of generating extreme examples, as originally proposed by Linden [6], performed so poorly and only outperformed the randomly selected suggestions by a narrow margin. This shows the importance of considering the preferences that are already known and those to be discovered in the design of suggestion strategies.

The simulations show that the simulated user is much more likely to state new preferences using the probabilistic strategy (statistically significant). Moreover, in the simulations the complete preference model was discovered up to 25 times more often with the probabilistic strategy than with randomly picked suggestions, up to 10 times more than using the extreme strategy, and 1.5 times more than the counting strategy. The probabilistic strategy has a better average performance than the attribute strategy.

Among the three model-based strategies, the probabilistic strategy provides the best results. However, it also makes the most assumptions about the preferences the user is likely to state. When these assumptions are not satisfied, the performance is likely to degrade. On the other hand, the counting strategy is the most robust among our strategies as it makes no assumptions whatsoever about the form of the user's preferences, while still achieving a large gain over simpler strategies. In the actual user studies, we decided to use the probabilistic strategy.

EXPERIMENTAL RESULTS: USER STUDY

In the user study, we are particularly interested in verifying:

Hypothesis 1: using model-based suggestions (at least the probabilistic strategy) leads to more complete preference models.

Hypothesis 2: using model-based suggestions leads to more accurate decisions.

Hypothesis 3: more complete preference models tend to give more accurate decisions, indicating that the reasoning underlying the model-based suggestions is correct.

We performed user studies using FlatFinder, a web application for finding student housing that uses real offers from a university database that was updated daily. The tool used the probabilistic strategy, as it was determined to be the best in the experiments with the simulated user. We recruited student subjects who had an interest in finding housing and thus were quite motivated for the task.

We studied two settings:

- in an unsupervised setting, we monitored user behavior on a publicly accessible example critiquing search tool for the listing. This allowed us to obtain data from over a hundred different users; however, it was not possible to judge decision accuracy since we were not able to interview the users themselves.
- in a supervised setting, we recruited 40 volunteer students use the tool under supervision. Here, we could determine decision accuracy because at the end we asked the subjects to carefully examine the entire database of offers to determine their target option. Thus, we could determine the switching rate and measure decision accuracy.

Each apartment comprises 10 attributes: the type of accommodation (room in a family house, room in a shared apartment, studio apartment, apartment), the rental price, the number of rooms, furnished (yes or no), the bathroom (private or shared), the type of kitchen (shared, private), the transportation available (none, bus, subway, commuter train), the distance to the university and the distance to the town center.

For numerical attributes, a preference consists of a relational operator (less than, equal, greater than), a threshold value and an importance weight between 1-5. For example, *price less than 600 Francs* with importance 4 indicates a relatively strong preference for an apartment below 600 Francs. For discrete attributes, a preference specifies a preferred value with a certain importance. Preferences are translated into numbers using standardized value functions and are combined by summing the results. The options are ordered so that the highest value is the most preferred.

The users stated a set of initial preferences and then obtained options by pressing the *search* button. Subsequently, they went through a sequence of *interaction cycles* where they could refine their preferences by critiquing the displayed examples. The system maintains their current set of preferences and the user could state additional preferences, change the reference value of existing preferences, or even remove one or more of the preferences. Finally, the process would finish with a user's final set of preferences, and a target choice chosen by the user from the displayed examples.

The search tool was made available in two versions:

- **C**, only showing a set of 6 candidate apartments without suggestions, and
- **C+S**, showing a set of 3 candidate apartments and 3 suggestions selected according to the probabilistic strategy

We now describe the results of the two experiments.

Online User Study

FlatFinder has been hosted on the laboratory web-server and made accessible to students looking for apartments during the winter of 2004-2005. For each user, it recorded anonymously a log of the interactions for later analysis. We set up a behavior so that users were alternatively presented with the versions with (**C+S**) and without (**C**) suggestions. We collected logs from 63 active users who went through several cycles of preference revision.

	tool **C**	tool **C+S**
number of critiquing cycles	2.89	3.00
number of initial preferences	2.39	2.23
number of final preferences	3.04	3.69
increment	0.65	1.46

Table 1. Average user behavior in the online experiment.

		Interaction	
	Characteristics (mean)	1st	2nd
group 1 (**C** first)	Decision Accuracy	0.45	0.80
	Preference Enumeration	5.30	6.15
group 2 (**C+S** first)	Decision Accuracy	0.72	0.67
	Preference Enumeration	5.44	4.50

Table 2. Results of the supervised user study. Decision accuracy and preference enumeration (the number of preferences stated) are higher when suggestions are provided.

In the following, whenever we present a hypothesis comparing users of the same group, we show its statistical significance using a paired student test. For all hypotheses comparing users of different groups, we use the unpaired student test to indicate statistical significance.

We first considered the increase from initial preference enumeration P_I to final preference enumeration P_F. This increment was on average 1.46 for the tool with suggestions **C+S** and only 0.64 for the tool **C**, showing the higher involvement of users when they see suggestions. This hypothesis was confirmed with $p = 0.002, t = -2.925$.

It is interesting to see that in both groups the users interacted for a similar number of cycles (average of 2.89 for **C** and 3.00 for **C+S** $p = 0.42$), and that the number of initial preferences is also close (average of 2.39 for **C** and 2.23 for **C+S** $p = 0.37$), meaning that the groups are relatively unbiased.

The result of the test shows clearly that users are more likely to state preferences when suggestions are present, thus verifying Hypothesis 1. They also show that model-based suggestions are significantly better than random ones. However, as this is an online experiment, we are not able to measure decision accuracy. Thus, we also conducted a supervised user study.

Supervised User Study

The supervised user study used the same tool as the online user study but users were followed during their interaction.

To measure improvement of accuracy, we instructed all of users to identify her most preferred item after she searched the database using interface 1. This choice was recorded and was called c_1. Then the users were instructed to interact with

the database using interface 2 and indicate a new choice (c_2) if the latter was an improvement on c_1 in their opinion. To evaluate whether the second choice was better than the initial one, we instructed the users to review all apartments (100 apartments in this case) and tell us whether c_1, c_2, or a completely different one truly seemed best.

Thus, the experiment allowed us to measure decision accuracy since we obtained the true target choice for each user. If users would stand by their first choice, it would indicate that they had found their target choice without further help from the second interface. If users would stand by their second choice, it would indicate that they had found their target choice with the help of the second interface. If users chose yet another item, it would indicate that they had not found their target choice even though they performed search with both interfaces.

40 (9 females) subjects of 9 different nationalities, mostly undergraduate students, took part in the study. Most of them (27 out of 40) had searched for an apartment in the area before and 26 out of 40 had used online tools to look for accommodations. Importantly, all subjects were motivated by the interest of finding a better apartment for themselves.

To overcome bias due to learning and fatigue, we divided the users in two groups, who were asked to interact with the versions in different order. Group 1 used tool **C** (interaction 1) and then **C+S** (interaction 2), while group 2 used the tools in the inverse order.

Both groups then went through the entire list to find the true most preferred option. For each version of the tool and each group, we recorded as decision accuracy as the fraction of subjects where the final choice made using that interface was equal to the target option. For both groups, we refer to the accuracy of interface 1 as a_1, and the accuracy of interface 2 as a_2.

We expected that the order of presenting the versions would be important: once they have realized their own preferences and found a satisfactory option, they are likely to be consistent with that; therefore we would have expected $a_2 > a_1$ in both cases. However we would expect that average accuracy would significantly increase with suggestions, and so we would see $a_2 >> a_1$ in the first group and a_2 only slightly higher than a_1 in group 2.

Decision Accuracy improves with suggestions

Figures 4 and 5 show the variation of decision accuracy for the two groups.

For group 1, after interaction with tool **C**, the accuracy is on average only 45%, but after interaction with **C+S**, the version with suggestions, it increases to 80%. This confirms the hypothesis that suggestions improve accuracy ($p = 0.00076, t = -2.6$). 10 of the 20 subjects in this group switched to another choice between the two versions, and 8 of them reported that the new choice was better. Clearly, the use of suggestions significantly improved decision accuracy for this group.

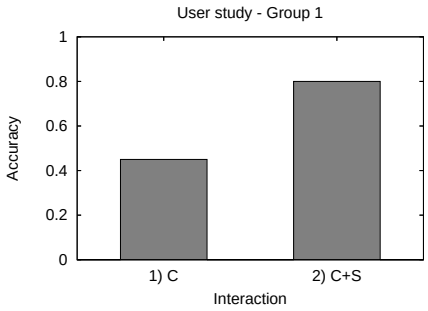

Figure 4. For group 1, accuracy dramatically increased when they used the version with suggestions (C+S) (p=0.00076).

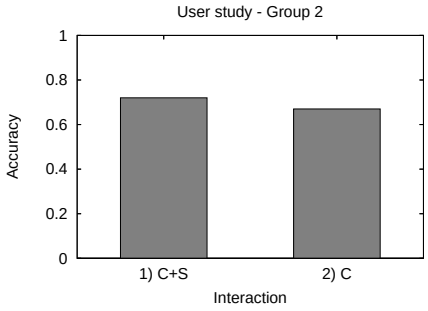

Figure 5. For group 2, accuracy was already very high when using the version with suggestions (C+S). Further interaction with the tool C (showing 6 candidates) did not increase accuracy any further. (p=0.33)

Users in group 2 used **C+S** directly and already achieved an average accuracy of 72%. We would have expected that a consequent use of tool **C** would have a small positive effect on the accuracy, but in reality the accuracy decreased to 67%. 10 subjects changed their final choice using the tool without suggestions and 6 of them said that the newly chosen was only equally good as the one they originally chose. The fact that accuracy does not drop significantly in this case is not surprising because users remember their preferences from using the tool with suggestions and will thus state them more accurately independently of the tool. We can conclude from this group that improved accuracy is not simply the result of performing the search a second time, but due to the provision of suggestions in the tool. Also, the closeness of the accuracy levels reached by both groups when using suggestions can be interpreted as confirmation of its significance.

We also note that users needed less cycles (and thus less effort) to make a decision with interface **C+S** (average of 4.15) than interface **C** (average of 5.92).

Interestingly, the price of the chosen apartment increased for the first group (average of 586.75 for **C** vs. 612.50 for **C+S**; $p = 0.04, t = -1.79$, statistically significant) whereas it de-

found	0.45	0.83
still not found	0.55	0.17
	$\Delta P <= 0$	$\Delta P > 0$

Table 3. For users who did not find their target in the first use of the tool, the table shows the fraction that did and did not find their target in the next try, depending on whether the size of their preference model did or did not increase.

creased for the second group (average of 527.20 for **C+S** to 477.25 for **C**; $p = 0.18$, the decrease is not statically significant). We believe that subjects in the first group did not find a good choice and thus paid a relatively high price to get an apartment with which they would feel comfortable. Conditioned by this high price they were then willing to spend even more as they discovered more interesting features through suggestions. On the other hand, subjects in group 2 already found a good choice in the first use of the tool, and were unwilling to accept a high price when they did not find a better choice in the second search without suggestions.

Thus, we conclude that Hypothesis 2 is confirmed: suggestions indeed increase decision accuracy.

Preference enumeration improves accuracy

In this study, we notice that when suggestions are present, users state a higher number of preferences (average of 5.8 preferences vs. only 4.8 without suggestions, $p = 0.021, t = 2.22$). Therefore, Hypothesis 1 is again confirmed.

To validate Hypothesis 3, that a higher preference enumeration also leads to more accurate decisions, we can compare the average size of preference model for those users who found their target solution with the first use of the tool and those who did not. In both groups, users who did find their target in the first try stated on average 5.56 preferences (5.56 in group 1 and 5.57 in group 2) while users who did not find their target stated only an average of 4.88 preferences (5.09 in group 1 and 4.67 in group 2). This shows that increased preference enumeration indeed improves accuracy, but this result was not statistically significant ($p = 0.17, t = -0.959$ overall). In fact, there is a chance that this correlation is due to some users being more informed and thus both making more accurate decisions and stating more preferences.

As an evaluation that is independent of user's a-priori knowledge, we only considered those users who did not find their target in the first try. As a measure of correlation of preference enumeration and accuracy, we considered how often an increase in preference enumeration in the second try led to finding the most preferred option on the second try. Table 3 shows that among users whose preference model did not grow in size, only 45% found their target. However, for those that increased their preference enumeration, 83% found their target as a result. Again, we see a good confirmation that higher preference enumeration leads to a more accurate decision with real users ($p = 0.04, t = 1.928$).

> 0	0.23	0.14	0.38
0	0.62	0.71	0.62
< 0	0.15	0.14	0.00
Δa,ΔP	< 0	$= 0$	> 0

Table 4. Variation of accuracy against variation of the number of stated preference P in the two steps of the user test. (Marginalization: each column sums to 1).

Finally, a third confirmation can be obtained by considering the influence that variations in the size of the preference model have on decision accuracy, shown in Table 4. Each column corresponds to users where the size of the preference model decreased, stayed the same, or increased, and shows the fraction for which the accuracy increased, stayed the same or decreased (note that when accuracy is 1 at the first step, it cannot further increase). We can see that a significant increase in accuracy occurs only when the size of the preference model increases; in all other cases there are some random variations but no major increases. The statistical test confirms the hypothesis that an increase in preference enumeration causes an increase in accuracy at a level of $p = 0.0322, t = 1.928$.

Thus, we conclude that hypothesis 3 is also validated by the user study: a more complete preference model indeed leads to more accurate decisions.

CONCLUSIONS

Search and recommender tools are an important part of computer usage today and present significant new human-computer interaction challenges that have been insufficiently addressed thus far. Among them is the problem of obtaining accurate user preferences through interaction.

Mixed-initiative systems such as example critiquing are a promising technology for efficiently eliciting accurate user preference models. Determining how to stimulate the user to state preferences on as many attributes as she may have is a key issue concerning such systems. We have developed a model for computing examples most suitable for stimulating preference expression and designed several suggestion strategies based on this model. The main principle is that suggestions should be options that are dominated under the current preference model but would no longer be dominated with the inclusion of additional preferences. In order to implement this principle with a minimum of assumptions about the user's preference model, we defined different strategies based on the concept of Pareto-optimality.

We first compared various suggestion strategies on simulations and determined the one that seemed to be the most effective. We confirmed its strong performance with live user studies, where we observed that the quality of the preference model, as measured by the number of stated preferences, increased almost twice as much with suggestions as without.

We followed this online user study by a supervised user study which also allowed us to measure decision accuracy. This study confirmed that the use of suggestions almost doubled

decision accuracy and allowed the search tool to find the most preferred option 80% of the time. This should greatly strengthen the performance of recommendation and search tools in applications ranging from decision support to e-commerce.

ACKNOWLEDGMENTS

The authors thank Vincent Schickel-Zuber for significant contribution in the development of the web based interface of FlatFinder and the anonymous reviewers for useful comments and suggestions.

REFERENCES

1. D. Bridge and A. Ferguson. Diverse Product Recommendations using an Expressive Language for Case Retrieval. In *Advances in Case-Based Reasoning*, Springer, 2002

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

3. R. Burke. Hybrid recommender systems: survey and experiments. *User Modeling and User-Adapted Interaction*, 12, 4 (2002), 331-370.

4. R. Burke, K. Hammond and E. Cooper. Knowledge-Based Navigation of Complex Information Spaces. In *Proceedings of the 13th National Conference on Artificial Intelligence*, AAAI press, 1996, pp. 462-468.

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

6. Greg Linden, Steve Hanks, and Neal Lesh. Interactive assessment of user preference models: The automated travel assistant. In *Proceedings, User Modeling '97*, 1997.

7. K. McCarthy, J. Reilly, L. McGinty and B. Smyth. Experiments in Dynamic Critiquing. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI'05)*, New York: ACM Press, 2005, pp. 175-182.

8. L. McGinty and B. Smyth. On the Role of Diversity in Conversational Recommender Systems. In *Proceedings of the 5th International Conference on Case-Based Reasoning* (ICCBR'03), 2003, pp. 276-290

9. David McSherry. Diversity-conscious retrieval. In *ECCBR*, pages 219–233, 2002.

10. David McSherry. Similarity and Compromise. *Proceedings of the 5th International Conference on Case-Based Reasoning*, LNAI 2689, Springer-Verlag, pp. 291-305, 2003

11. J.W. Payne, J.R. Bettman, and E.J. Johnson. *The Adaptive Decision Maker*. Cambridge Univ. Press, 1993.

12. R. Price and P.R. Messinger. Optimal Recommendation Sets: Covering Uncertainty over User Preferences. In *Proceedings of the 20th National Conference on Artificial Intelligence* (AAAI'05), 2005, pp. 541-548.

13. Pearl Pu and Boi 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.

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

15. Pearl Pu, Boi Faltings and Marc Torrens, Effective Interaction Principles for Online Product Search Environments. In *Proceedings of the IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology and Web Intelligence*, 2004, pp. 724-727

16. Pearl Pu and P. Kumar, Evaluating Example-Based Search Tools. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC'04)*, ACM Press, 2004, pp. 208-217.

17. Pearl Pu and Li Chen, Integrating Tradeoff Support in Product Search Tools for E-Commerce Sites. In *Proceeding of the 6th ACM Conference on Electronic Commerce* (EC'05), ACM Press, 2005, pp. 269-278.

18. James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Dynamic critiquing. In *ECCBR*, pages 763–777, 2004.

19. Hideo Shimazu. Expertclerk: Navigating shoppers buying process with the combination of asking and proposing. In *Proceedings of the 17 International Joint Conference on Artificial Intelligence (IJCAI'01)*, volume 2, pages 1443–1448, 2001.

20. Sybil Shearin and Henry Lieberman. Intelligent profiling by example. In *Intelligent User Interfaces*, pages 145–151, 2001.

21. Barry Smyth and Lorraine McGinty. The power of suggestion. In *IJCAI*, pages 127–132, 2003.

22. B. Smyth, P. McClave. Similarity vs. Diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR'01)*, Springer-Verlag, 2001, pp. 347-361.

23. A. Tversky, S. Sattath, and P. Slovic. Contingent weighting in judgment and choice. *Psychological Review*, 95:371–384, 1988.

24. C.N. Ziegler, S.M. McNee, J.A. Konstan, G. Lausen. Improving Recommendation Lists Through Topic Diversification. In *Proceedings of the 14th International World Wide Web Conference* (WWW'05), 2005, pp. 22-32.

25. M.D. Williams and F.T. Tou. RABBIT: An Interface for Database Access. In *Proceedings of the ACM '82 Conference*, ACM Press, 1982, pp. 83-87.