Conversational Recommenders with Adaptive Suggestions

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

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

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

ABSTRACT

We consider a conversational recommender system based on example-critiquing where some recommendations are suggestions aimed at stimulating preference expression to acquire an accurate preference model. User studies show that suggestions are particularly effective when they present additional opportunities to the user according to the look-ahead principle [32].

This paper proposes a strategy for producing suggestions that exploits prior knowledge of preference distributions and can adapt relative to users' reactions to the displayed examples.

We evaluate the approach with simulations using data acquired by previous interactions with real users. In two different settings, we measured the effects of prior knowledge and adaptation strategies with satisfactory results.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User Machine Systems—*human factors, software psychology*

General Terms

Human factors

Keywords

Recommender systems, example-critiquing interfaces, personalized search

1. 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 [4, 3, 5, 20, 24]. 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, which have been poorly addressed so far, but are key to the user success rate of such systems on e-commerce sites.

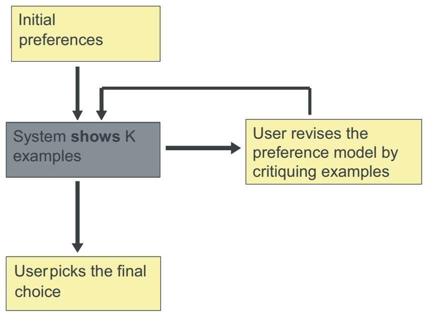

Initial preferences
System **shows** K examples
User revises the preference model by critiquing examples
User picks the final choice

Figure 1: : The generic system-user interaction model of a preference-based search and recommender system.

Utility theory provides a solid mathematical foundation for recommendations [11]. 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 [17]. To allow such construction to take place, users must be able to explore the space of possible options while building their preference model.

2. EXAMPLE-CRITIQUING

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 [30] and works by showing k examples to a user in each interaction cycle. Showing a set of examples allows the system to correct for an incomplete or inaccurate model of the user's preferences [9]. 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 [21]. It has been used in various product search and recommender tools [14, 5, 26, 20, 28, 25].

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.

Example critiquing was first proposed in [30] and has since been used in several recommender systems, such as FindMe [5], ATA [14], SmartClient [20], ExpertClerk [26] and the system of Shearin & Lieberman [25]. The ATA system [14] 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 [22] 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 [19] showed that the implementation of tradeoff support can increase users' decision accuracy by up to 57%.

In *dynamic critiquing* [24], a popular family of example critiquing interfaces, a metaphor of navigation through the product space is implemented; the interface proposes pre-computed critiques (simple and compound) that can be selected by the users. McCarthy et al. [15] showed that users who more frequently applied compound critiques in a critiquing interface were able to reduce interaction cycle from 22 to 6 .

2.1 Example-critiquing with suggestions

We consider an example-critiquing framework where

1. Preferences are stated as reactions to displayed options (as "The price should be less than 600"). Such critiques are user-motivated.
2. These critiques are used as feedback in the next interaction cycle to generate a new set of displayed items.

If certain preferences are missing from the current model of the user, the system may provide solutions that do not satisfy those unknown preferences. If the user is aware of all of her preferences, she realizes the necessity to state them to the system. However this is not usually the case, because the user might not know all the available options. Moreover, stating a preference costs some user effort and she would make that effort only if she perceives this as beneficial.

The influence of current examples is known as the *anchoring effect* [7]. To enable the user to refocus the search in another direction, many researchers have suggested to display alternatives or diverse examples, in addition to the best options (candidates). In one user study it was observed that the majority of critiques (79%) were a reaction to seeing an additional opportunity rather than seeing unsatisfactory examples [32].

Different strategies for suggestions have been proposed in the literature. Linden [14] used extreme examples, where some attribute takes an extreme value. Others use diverse examples as suggestions [27, 28, 26].

However, an extreme example might often be an unreasonable choice: it could be a cheap flight that leaves in the early morning, a student accommodation where the student has to work for the family, an apartment extremely far from the city. Moreover, in problems with many attributes, there will be too many extreme or diverse examples to choose from, while we have to limit the display of examples to few of them.

The user should be motivated to state new preferences by options that are reasonable choices (given the previously stated preferences) and have a potential of optimality (a new preference is required to make them optimal).

This was expressed in the *lookahead principle* [23]:

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

Model-based suggestions are calculated based on that principle. Results show that such suggestions are highly attractive to users and can stimulate them to express more preferences to improve the chance of identifying their most preferred item by up to 78% [33].

The lookahead principle was implemented by considering Pareto-optimality: suggestions are evaluated according to their probability of becoming Pareto-optimal. The advantage of this method is that it is qualitative, since it does not rely on any particular parametrization: an option is Pareto-optimal if there is no other option that is better or equally preferred with respect to all the preferences and strictly better for at least one preference.

To become Pareto-optimal, the new preference has to make the current solution escape the dominance with better solutions (the "dominators").

An example of a conversational recommendation system is FlatFinder, a tool for finding student accommodation using example critiquing, using data available from the faculty housing program (containing around 200 items). Figure 2 shows an example of an interaction with FlatFinder. The user has posted preferences for a cheap accommodation (price less than 450) with a private bathroom. The tool shows 3 best options, which are all rooms in a family, with bus or no public transport at all. It also displays 3 suggestions, which show that paying a bit more, the user can have a studio or a small private apartment where both are closer to the university and the centre of town; the second also is close to a metro station. The last option is still a room but is closer to the center than the options currently showed as best options. Seeing the suggestions, the user could realize that she cares about these features, and then state a preference on the transportation, for the accommodation type or on distance from centre and university. We have used FlatFinder in extensive user studies reported in [23, 31, 33].

2.2 Adaptive preference elicitation

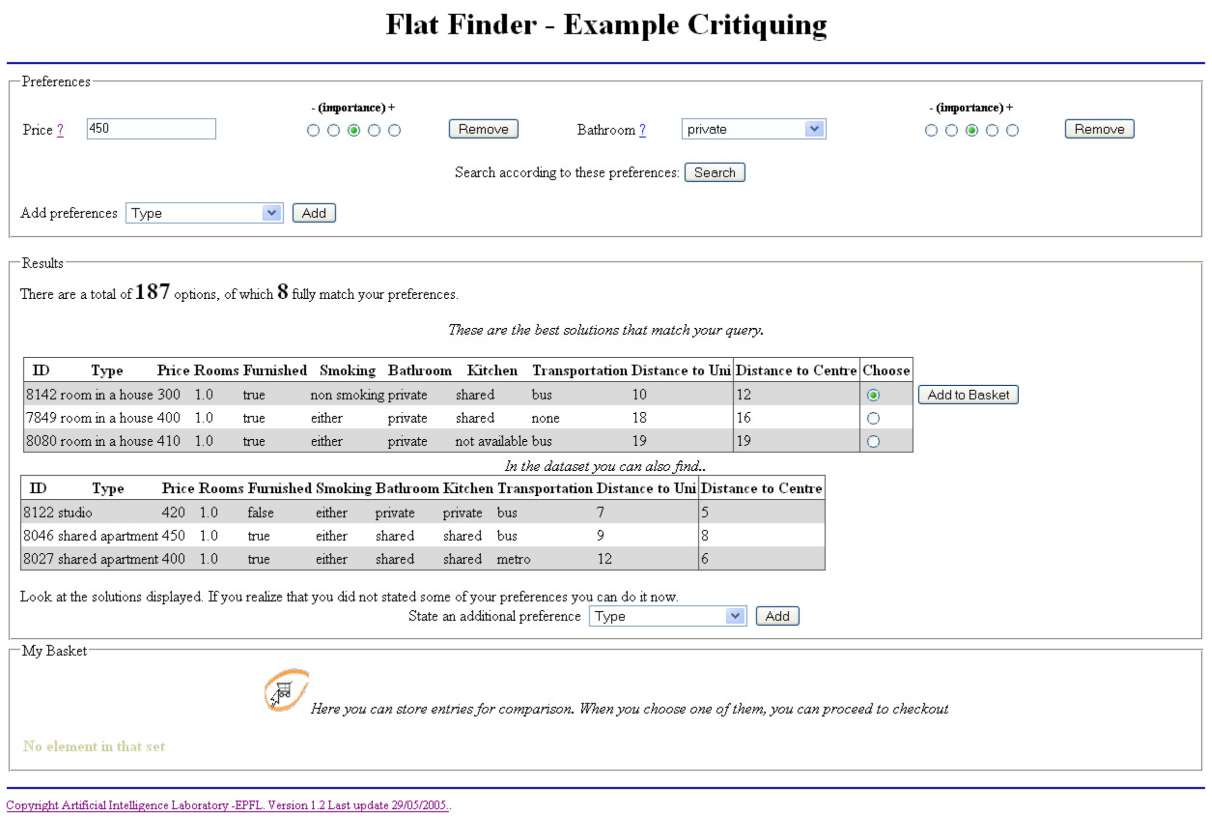
In artificial intelligence, several people have dealt with the problem of understanding the behavior of an autonomous agent on the basis of observations that his next actions can be anticipated [2, 10]. For instance, Lang [13] considered a qualitative approach for inferring the agent's goal and his plausible next actions under the assumption that agents are goal-seeking and have a given preference relation over plans of actions.

In the context of a preference-based search problem, the acquisition of the user model might be facilitated if the system could anticipate the behavior of the the user given observations of their past behavior. Recently, several researchers followed this intuition by considering an adaptive strategy for utility-based elicitation and question/answer interface.

Chajewska et al. [6] proposed an elicitation procedure that models the uncertainty of the utility function and selects assessment questions to maximize the expected value of information. Boutilier [1] has extended this work by taking into account values of future questions to further optimize decision quality while minimizing user effort. The elicitation procedure itself is a optimized based on a partially observable Markov decision process(POMDP) where the goal is to minimize the user effort.

Stolze considers how to optimize a an online question/answer interface, by adapting the questions taking into account the distribution of possible answers and solving a decision tree optimization problem [29].

Other works focused on how to improve decision aid tools using prior knowledge. Price and Messinger in [18] optimize the set of examples given an expectation of the user's preferences, without actually asking the users to state their own preferences.



Flat Finder - Example Critiquing

Preferences

Price ? 450 - (importance) + ○ ○ ● ○ ○ Remove Bathroom ? private - (importance) + ○ ○ ● ○ ○ Remove

Search according to these preferences: Search

Add preferences Type Add

Results

There are a total of **187** options, of which **8** fully match your preferences.

These are the best solutions that match your query.

ID	Type	Price	Rooms	Furnished	Smoking	Bathroom	Kitchen	Transportation	Distance to Uni	Distance to Centre	Choose
8142	room in a house	300	1.0	true	non smoking	private	shared	bus	10	12	●
7849	room in a house	400	1.0	true	either	private	shared	none	18	16	○
8080	room in a house	410	1.0	true	either	private	not available	bus	19	19	○

Add to Basket

In the dataset you can also find..

ID	Type	Price	Rooms	Furnished	Smoking	Bathroom	Kitchen	Transportation	Distance to Uni	Distance to Centre
8122	studio	420	1.0	false	either	private	private	bus	7	5
8046	shared apartment	450	1.0	true	either	shared	shared	bus	9	8
8027	shared apartment	400	1.0	true	either	shared	shared	metro	12	6

Look at the solutions displayed. If you realize that you did not stated some of your preferences you can do it now.

State an additional preference Type Add

My Basket

Here you can store entries for comparison. When you choose one of them, you can proceed to checkout

No element in that set

Copyright Artificial Intelligence Laboratory -EPFL. Version 1.2 Last update 29/05/2005..

Figure 2: *The FlatFinder example-critiquing interface.*

2.3 Contribution of this paper

In this paper we consider a mixed initiative tool for preference-based search in which preferences are stated as critiques to shown examples (user-motivated critiques) and suggestions are presented to stimulate preference expression.

We consider how suggestions can profit from prior knowledge and propose a strategy for adapting the suggestions based on the user's reactions. The interaction cycle of a conversational recommender system with adaptive suggestions is shown in Figure 3.

As far as we know, we are the first to propose an adaptive strategy of preference acquisition in a conversational recommender system based on example-critiquing.

3. THE FRAMEWORK

3.1 Basics

We assume that O is the set of options $o_1, .., o_n$ defined over A, set of attributes $\{a_1, .., a_n\}$ of domains $D_1, .., D_m$; $a_i(o)$ is the value that o takes on attribute a_i.

Domains can be qualitative or numeric. A **qualitative domain** (such as colors or names) consists of an enumerated set of possibilities; a **numeric domain** has numerical values (as price, distance to center), either discrete or continuous. For numeric domains, we consider a function $range(Att)$ that gives the range on which the attribute domain is defined. We define qualitative (respectively numeric) attributes those with qualitative (numeric) domains.

Preferences are stated on individual attributes. A preference r_i applies to a particular attribute a_i and results in a total or partial order on the values in the domain D_i of a_i.

A preference model R consists of a set of preferences. We assume that a preference r_i is expressed by a cost function c_i. Since a preference always applies to the same attribute we can use $c_i(o)$ instead of $c_i(a_i(o))$.

We assume that the cost 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 $c_i(s_1) < c_i(s_2)$.

The individual costs obtained by each preference are merged with a particular combination function, for example a weighted sum.

$$C(o) = \sum_i w_i * c_i(o) \tag{1}$$

The *candidate* best options can be found by sorting the database items according to their cost. This is known as the *top-k query* [8]. The set of options retrieved $\{o_1, .., o_k\}$ is such that $C(o_1) \leq C(o_2) \leq .. \leq C(o_k)$ and for any other option $\bar{o}$ in the database $C(\bar{o}) \geq C(o_k)$.

3.2 Pareto-dominance and optimality

We model preferences by standardized functions that correctly reflect the preference order of individual attribute values but may be numerically inaccurate.

Pareto-optimality is the strongest concept that can be applied without knowledge of the numerical details of the penalty functions.

An option o is **(Pareto) dominated by** another option $\bar{o}$ (equivalently we say that $\bar{o}$ dominates o) if

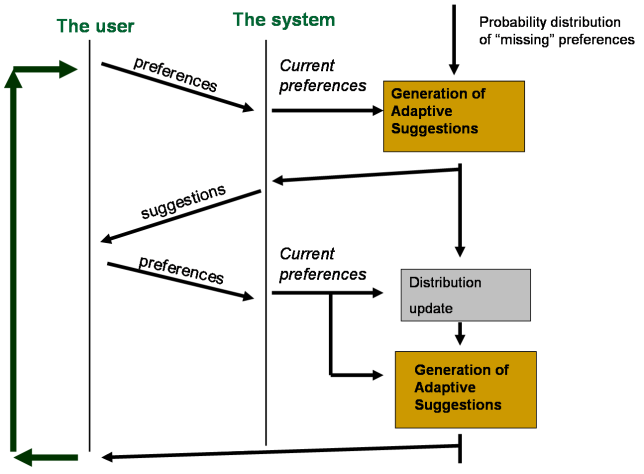

The user
The system
Probability distribution of "missing" preferences
preferences
Current preferences
Generation of Adaptive Suggestions
suggestions
preferences
Current preferences
Distribution update
Generation of Adaptive Suggestions

Figure 3: With adaptive suggestions, the tool learns from the user by observing the reaction to displayed examples. For instance, if we have shown to the user an option with value `subway` **for attribute "transportation" and she has not stated any critique about the kind of transportation, the probability that the user has a preference for subway as transportation will decrease. In the next interaction cycle, the system is likely to choose options with a different value.**

- $\bar{o}$ is not worse than o according to all preferences in the preference model: $\forall c_i \in R : c_i(\bar{o}) \leq c_i(o)$
- $\bar{o}$ is strictly better than o for at least one preference: $\exists c_j \in R : c_j(\bar{o}) < c_j(o)$

An option o is **Pareto-optimal** if it is not Pareto-dominated by any other option.

The **dominating set** $O^+(o)$ is the set of options that dominates the option o. The **equal set** $O^=(o)$ is the set of options that are equally preferred with o.

PROPOSITION 1. *A dominated option o with respect to R becomes Pareto-optimal with respect to $R \cup r_i$ if and only if o is*

- *strictly better with respect to r_i than all options that dominate it with respect to R and*
- *not worse with respect to r_i than all options that are equally preferred with respect to R.*

Thus, the dominating set $O^>$ and the equal set $O^=$ of a given option are the potential dominators when a new preference is considered.

3.3 Assumptions about the cost functions

We suppose that the system knows that each user has preference value functions in a known parameterized family. Here we assume a single parameter θ, but the method can be generalized to handle cases of multiple parameters.

$$\mathbf{c}_i = c_i(\theta, a_i(o)) = c_i(\theta, o) \tag{2}$$

We use $r_{i,\bar{\theta}}$ to denote the preference on attribute i with parameter $\bar{\theta}$, corresponding to the cost function $c_i(\bar{\theta}, o)$.

For simplicity, we let the parameter θ represent the *reference point* of the preference stated by the user.

Consider a preference statement like "I prefer German cars" , meaning that cars manufactured in Germany are preferred to cars manufactured in another country. It can be represented by the following cost function c_i considering the attribute a_i being the country and $\theta =$ german.

$$c_i(\theta, x) \equiv \textbf{ if } a_i(x) = \theta \textbf{ then } 0 \textbf{ else } 1. \tag{3}$$

For numeric domains we consider that the user can choose between a fixed set of qualitative statements like:

- `LowerThan(`θ`)`: the value should be lower than θ
- `Around(`θ`)`: the value should be around θ
- `GreaterThan(`θ`)`: the value should be greater than θ

We suppose that for a given implementation we will choose a particular form of representing these statements. The designers of the application will consider the kind of preference statements that the user can state through the user interface and the way to translate the statements into quantitative cost functions. A similar approach is taken by Kiessling in the design of `PREFERENCE SQL` [12], a database system for processing queries with preferences.

A reasonable possibility is to represent their cost functions as graded step functions, with the penalty value increasing from 0 to 1. In case where the degree of violation can worsen indefinitely, a ramp function can be used. For example, in a system for flight reservations where the user states that she prefers to arrive before a certain hour (θ), a possible function is the following:

$$c_i(\theta, x) = \begin{cases} (x - \theta) & \textbf{if } a_i(x) > \theta \\ 0 & \textbf{otherwise} \end{cases} \tag{4}$$

4. PRIOR DISTRIBUTION

We suppose to have a prior probability distribution over the possible preference models, learnt from previous interactions with the system. We consider

- p_{a_i} the probability that the user has a preference over an attribute,
- $p_i(\theta)$ the distribution of probability over the parametric value of the preference representation for the cost function c_i,
- $p(r_{i,\theta})$ the probability that the user has a preference on attribute i and its parameter is θ. We assume it to be $p_{a_i} * p_i(\theta)$.

Such distribution will be updated by the user during the interaction based on the user's action.

5. ADAPTIVE SUGGESTIONS

According to our look-ahead principle, we choose suggestions that have the highest likelihood of becoming optimal when a new preference is added. Since we would like to avoid sensitivity to the numerical error of the cost functions, we use the concept of Pareto-optimality.

In this section we show how to compute the probability that a given option becomes Pareto optimal (breaks all dominance relations with options in its dominating set). These formulas depend on the probability distributions p_{a_i} and $p_i(\theta)$ that we have introduced before. At the beginning of the interaction, they are initialized by prior knowledge, considering the preference models of previous users. During the interaction, they are updated according to the observations of the user's actions.

5.1 Probability of escaping attribute dominance

We first consider the probability of breaking one dominance relation alone. To escape dominance, the option has to be better than its dominator with respect to the new preference.

The probability that a preference on attribute i makes o_1 be preferred to o_2 can be computed integrating over the values of θ for which the cost of o_1 is less than o_2.

$$\delta_i(o_1, o_2) = \int_\theta H(c_i(\theta, o_2) - c_i(\theta, o_1))p(\theta)d\theta \qquad (5)$$

where H is the function $H(x) \equiv$ **if** $(x > 0)$ **then** 1 **else** 0.

For a qualitative domain, we iterate over θ and sum up the probability contribution of the cases in which the value of θ makes o_1 preferred over o_2.

For breaking the dominance relation with all the options in the dominating set when a new preference is set on attribute a_i, all dominating options must have a less preferred value for a_i than that of the considered option. For numeric domains, we have to integrate over all possible values of θ, check whether the given option o has lower cost and weigh the probability of that particular value of θ.

$$\delta_i(o, O^{\geq}) = \int [\prod_{o' \in O^{>}} H(c_i(\theta, o') - c_i(\theta, o))$$
$$\prod_{o'' \in O^{=}} H^*(c_i(\theta, o'') - c_i(\theta, o))]p(\theta)d\theta$$

where H^* is a modified Heaviside function that assigns value 1 whenever the difference of the two costs is 0 or greater. ($H^*(x) \equiv$ **if** $(x \geq 0)$ **then** 1 **else** 0).

For qualitative domains, simply replace the integral with a summation over θ.

5.1.1 Example

In general the cost functions in use for a particular system can lead to substantial simplification of the calculations of the integrals. Considering the cost function

$$c_i(\theta, x) = \textbf{ if } a_i(x) = \theta \textbf{ then } 0 \textbf{ else } 1$$

the probability of breaking a dominance relation between option o_1 and o_2 simplifies to the probability that the value of option o_1 for attribute i is the preferred value, when it differs from the value of o_2.

$$\delta_i(o_1, o_2) = \begin{cases} p(\theta = a_i(o_1)) & \textbf{if } a_i(o_1) \neq a_i(o_2) \\ 0 & \textbf{otherwise} \end{cases} \qquad (6)$$

5.2 Probability of becoming Pareto-optimal

To make an interesting suggestion, it is sufficient that an option has a chance of becoming Pareto-optimal, i.e. escape dominance in at least one of the attributes in the set A of all attributes modeling the options. This is given by the combination of the events that the user has a preference on a particular attribute a_i, and the event that this preference will break all attribute dominance in that attribute:

$$p^{opt}(o) = 1 - \prod_{a_i \in A} (1 - p_{a_i}\delta_i(o, O^{\geq}(o))) \qquad (7)$$

where $\delta_i(o, O^{\geq})$ is the probability of simultaneously escaping many dominance relations and p_{a_i}, as said before, is the probability that the user has a preference over attribute a_i.

To generate suggestions that are adapted to the user, we update the value p_{a_i} according to the user's actions. The computation of δ_i depends on $p_i(\theta)$, that is also updated according to the user's behavior.

Following the lookahead principle, the options for which p^{opt} is greatest are chosen as the best suggestions

5.3 Belief Update

Model-based suggestions can become very effective if they can adapt to the user, responding to his actions. In particular, after the user has made a query and some example options have been shown (*candidates* and *suggestions*), the user might or might not state an additional preference. Observing the reaction of the user to the examples, the system can refine the uncertainty over the preference model. Suggestions stimulate the expression of those preferences on the values that are shown; therefore, if the user does not state any preference, the likelihood that there is a preference on those values will decrease.

After each interaction cycle, we need to update the p_{a_i} and $p(\theta_i)$ for all attributes i, on the basis of the user's reaction to the shown example.

We suppose to know a model of the user's reaction behavior, composed of the following probabilities (that refer to the situation in which at least one relevant option is displayed).

- $p(st|r_{i,\theta})$ the probability that the user **states** a preference given that the preference **is** in the user's preference model
- $p(st|\neg r_{i,\theta})$ the probability that the user **states** a preference given that the preference **is not** in the user's model

We will expect the second to be relatively small: from our experience in the user tests, users of example-based tools state preferences only when these are relevant (in contrast to the form-filling approach, where users are more likely to state preferences they do not have).

We use the Bayesian rule to update the probability of a particular preference being present in the user model, in the condition that a relevant option is presented to the user in the set of displayed examples. The probability that a preference on attribute i with parameter θ is present when the user has stated or not stated any critique is the following:

$$\begin{cases} p(r_{i,\theta}|st) = \dfrac{p(st|r_{i,\theta})p(r_{i,\theta})}{p(st|r_{i,\theta})p(r_{i,\theta}) + p(st|\neg r_{i,\theta})p(\neg r_{i,\theta})} \\ p(r_{i,\theta}|\neg st) = \dfrac{p(\neg st|r_{i,\theta})p(r_{i,\theta}}{p(\neg st|r_{i,\theta})p(r_{i,\theta}) + p(\neg st|\neg r_{i,\theta})p(\neg r_{i,\theta})} \end{cases} \qquad (8)$$

where $p(\neg r) = 1 - p(r)$.

Once we know the new value for $p(r_{i,\theta})$, we can compute the new values for p_{a_i} and $p_i(\theta)$ that will be used to generate suggestions at the next interaction cycle.

$$p_{a_i} = \int_\theta p(r_{i,\theta})d\theta \qquad (9)$$

$$p_i(\theta) = p(r_{i,\theta}) * (1/p_{a_i}) \qquad (10)$$

The new value for p_{a_i}, the probability that there is a preference over a particular attribute, is obtained by the cumulative addition of the joint probability $p(r_{i,\theta}$ over $\theta)$, and the the parametric distribution $p_i(\theta)$ by dividing the joint probability $p(r_{i,\theta})$ by the attribute probability p_{a_i}.

5.4 Algorithm for belief update

Thanks to our simplification in the type of cost functions, the algorithm for updating the probability distribution is not particularly complicated. The parameter θ represents the "reference value" of the preference (the most preferred value for qualitative domains; the extreme of acceptable values for numeric domains).

For each of the displayed options o and for each of the attributes a_i, we update the probability $p(r_{i,a_i(o)})$ (where $\theta = a_i(o)$) conditioned on whether the user has stated an additional preference on i or not, as we have shown in the previous paragraph.

The algorithm is shown in Algorithm 1, where `update` refers to the probability update performed by Equation 8, 9 and 10.

Algorithm 1: **Belief update**
`newPref` contains the indexes of attributes on which a preference has been stated in the last interaction.

```
for all option $o \in$ DISPLAYED-OPTIONS do
    for all attribute $a_i \in$ ATTRIBUTES do
        $\theta = a_i(o)$
        stated = ($i \in$ newPref)
        update($i$, $\theta$, stated)
```

6. EVALUATION

In previous work, we carried out user studies to validate the effectiveness of example-based tools for preference-based search, with and without suggestions ([23, 33]). In these experiments, model-based suggestions were calculated without prior knowledge (thus, assuming constant probability distribution) and no belief update.

We measured decision accuracy. 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. We obtained the strongest increase in accuracy, to 70%, when using example-critiquing with suggestions.

In the next section, we evaluate the performance of adaptive suggestions using prior knowledge and Bayes reasoning.

6.1 Validation of adaptive suggestions with simulations

We evaluated our adaptive strategy of generating suggestions with simulations. Assuming that our look-ahead strategy is correct, we look at the effectiveness of the suggestions in stimulating preference expression. The simulated user behaves according to an opportunistic model by stating a preference whenever the suggestions contain an option that would become optimal if that preference was added to the model with the proper weight.

To obtain realistic prior distributions we considered logs from previous user studies using the FlatFinder tool. A total of 100 interaction logs, each constituting a complete search process, were considered.

For the adaptive suggestions, we need an estimate for the probability that the user states a preference given that an option is shown (in the Bayesian update formula, Equation 8). We substitute this with the value of the estimated probability of becoming pareto optimal, according to our look-ahead principle.

6.2 The effect of prior knowledge

We run simulations in two settings. In the first, we used the logs to draw random preference models and measure the percentage of time that the look-ahead principle is satisfied (an option that is selected by the system as a suggestion becomes Pareto optimal). We call this measure the *hit rate*. We compare different strategies of suggestions: model-based suggestions using prior knowledge, model-based suggestions assuming uniform distribution, *extremes* strategy (suggesting options where attributes take extreme values, as proposed in [14]), the *diversity* strategy (computing the 20 best solutions according to the current model and then generating a maximally diverse set of 5 of them, following the proposal of [16]).

	hit-rate
model based suggestions + prior knowledge	87%
model based suggestions	61%
diversity strategy	25%
extreme strategy	13%

Table 1: The average number of discovered preferences. We compare the model-based suggestions with prior knowledge, the standard model-based suggestion strategy assuming uniform distribution, the strategy of maximizing diversity and the extreme strategy.

Considering the logs of previous interactions with the tool, we could set up the value for $p(r_{i,\theta})$, the probability that the user has a preference on attribute i and its parameter is θ, assigned to the occurrence of the preference $r_{i,\theta})$.

The results (Table 1) show that prior knowledge of the distribution of the preferences can greatly help in making better suggestions (87% against 61%).

6.3 Simulating a real user

To better evaluate suggestions in a more realistic scenario, we considered the simulation of an interaction between a simulated user and the example-critiquing tool.

In the simulations, we set $p(st|\neg r_{\mathrm{i},\theta}) = 0$, meaning that the user states only preferences that she really has. This is reasonable because the user can only state preferences on her own initiative. Therefore, $p(\neg st|\neg r_{\mathrm{i},\theta}) = 1$.

We need to define $p(st|r_{\mathrm{i},\theta})$. As recalled before, in a user study it was observed that the majority of critiques (79%) were a reaction to seeing an additional opportunity rather than seeing unsatisfactory examples [32], providing evidences for the correctness of the lookahead principle. Therefore, we assume a probabilistic model of the user coherent with the lookahead principle

$$p(st|r_{\mathrm{i},\theta}) = p^{opt}(o) \qquad (11)$$

where o is the option displayed and $p^{opt}(o)$ the estimated probability of optimality. We think that this is a cautious approach: in many real situations the user will state a preference when a new opportunity is shown, even if optimality is not achieved.

Given these assumptions, any time that an option is shown and the user does not critique a given preference, the new value for the probability that the preference is present $p^{new}(r_{i,\theta}) = p(r_{\mathrm{i},\theta}|\neg st)$ can be written as:

$$\begin{aligned} p^{new}(r_{i,\theta}) = & \quad p(r_{i,\theta}) \frac{1 - p^{opt}(o)}{[1 - p^{opt}(o)]p(r_{i,\theta}) + [1 - p(r_{i,\theta})]} \\ = & \quad \frac{p(r_{i,\theta}) - p^{opt}(o)p(r_{i,\theta})}{1 - p^{opt}(o)p(r_{i,\theta})} \end{aligned}$$

The belief update depends on $p^{opt}(o)$, that is an estimation of the quality of the suggestion. This means that if the system shows an option that has no chance of becoming optimal ($p^{opt}(o) = 0$) then $p(r)$ will remain unchanged. Instead if $p^{opt}(o) = 1$ (an ideal suggestion is shown), $p(r)$ will go to 0 if no reaction is observed.

To handle the probability distribution of continuous attributes, we discretized the domains in few intervals, because otherwise the probabilistic update would be untractable.

We split the data (100 logs of user interactions) in two sets.

1. A learning data-set, used to represent prior knowledge to feed the the adaptive tool.
2. A test data-set, used to generate preference models of simulated users.

We ran several simulations with a random split of samples between the learning and the test data-set.

In the simulations, users have a set of preferences generated according to a particular probabilistic distribution. Every step of the simulation represents an interaction with the recommender system. Based on the preferences that are known to the system a set of candidates and suggestions are retrieved at each step. When the suggestions retrieved by the strategy satisfies the look-ahead principle, a new preference is stated and considered in the user model. In this case, we say that a new preference is discovered and the suggestions were appropriately chosen.

According to our user studies, on average the users interact on average for 6.25, of which in 3.30 cycles the users did not either remove nor add preferences. We implemented the simulation so that the interaction continues until either the user model is complete or the simulated user states no further preference twice in a row (choosing a prudent approach).

If all preferences are stated, the preference model is complete (a *full discovery* of the preferences). In this case, it is guaranteed that the user can find the target (his true best option) by a series of tradeoff actions that modify the relative importance of the preferences.

The results are shown in Table 2. We compare the basic formulation of model-based suggestions (assuming uniform distribution), model-based suggestions with prior knowledge and adaptive suggestions. We measure the fraction of preferences discovered.

Even with a simple probabilistic model of the user, the gains obtained by adaptive suggestions become considerable, specially when 3 or more suggestions are shown (72% of runs achieve full discovery for adaptive suggestions, 58% prior distribution, 42% normal suggestion). There is a certain gain of adaptive suggestions, that are better than suggestions generated according to prior distribution. It is important to note that the performance of suggestions with prior knowledge degrades with respect to the first simulation setting.

It would be interesting to compare the effects of different choices for the value $p(st|r_{1,\theta})$ of the probabilistic behavior of the user.

	number of suggestions shown			
hit-rate	1	2	3	4
adaptive suggestions	39.7%	61.7%	81.0%	97.4%
suggestions + prior knowledge	39.7%	54.8%	72.4%	94.0%
basic suggestions	32.7%	52.0%	50.2%	65.8%

full discovery	1	2	3	4
adaptive suggestions	32%	53%	72%	88%
suggestions + prior knowledge	32%	46%	58%	85%
basic suggestions	30%	44%	42%	53%

Table 2: The average fraction of discovered preferences and the percentage of full discovery (interactions that acquire a complete preference model) in the simulation with probabilistic profile acquired from real logs. We compare the adaptive model-based suggestions, model-based suggestions with prior knowledge and the standard model-based suggestion strategy assuming uniform distribution.

7. CONCLUSION

The internet allows access to an unprecedented variety of information, but forces people to see the world through the restricted lens of a computer. Currently, only a small minority of users manage to reliably find the items they are looking for, and better recommendation tools are badly needed.

We believe that the mixed-initiative approach for preference-based search presented in this paper is a major step in this direction. Such tools model a user's preferences and show a set of options generated on the basis of this model. The user reacts by either picking one as her choice or modifying her preference model. To increase the quality of the final decisions, the system provides suggestions to stimulate preference expression. The main contribution of this paper is to further increase the recommendation accuracy by refining the uncertainty over users' preferences and making suggestions that are adaptive to the users reactions.

We evaluated the effectiveness of suggestion strategies in eliciting user preferences. We divide the available data logs of user interactions in two parts: one is used for learning and the other for testing the strategy. We showed that our approach works significantly better than standard model-based suggestions. While a considerable part of the improvement is due to prior knowledge, the adaptation of suggestions according to the reactions gives an important improvement, especially in cycles in which the user does not state additional preferences.

In the simulations, we manage to discover the complete set of preferences almost all the time, a dramatic improvement from earlier results using suggestions with fixed probabilities. In earlier work, we have shown that suggestions with fixed probabilities already increase decision accuracy from 25% to 70%, so we would expect an even higher accuracy from adaptive suggestions, thus bringing us closer to our goal of a practically reliable conversational recommendation system. We are hoping to verify this expectation through additional user studies soon.

8. REFERENCES

[1] C. Boutilier. A pomdp formulation of preference elicitation problems. In *ĺProceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI'02)*, pages 239–246, 2002.

[2] R. I. Brafman and M. Tennenholtz. Modeling agents as qualitative decision makers. *Artif. Intell.*, 94(1-2):217–268, 1997.

[3] R. D. Burke. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, 12(4):331–370, 2002.

[4] R. D. Burke, K. J. Hammond, and B. C. Young. Knowledge-based navigation of complex information spaces. In *AAAI/IAAI, Vol. 1*, pages 462–468, 1996.

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

[6] U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of*

the Seventeenth National Conference on Artificial Intelligence (AAAI'00)*, pages 363–369. AAAI Press / The MIT Press, 2000.

[7] A. T. Daniel Kahneman, Paul Slovic. Judgement under uncertainity: Heuristics and biases. *Science*, 185:1124–1131, 1974.

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

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

[10] P. Gorniak and D. Poole. Predicting future user actions by observing unmodified applications. In *AAAI/IAAI*, pages 217–222. AAAI Press / The MIT Press, 2000.

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

[12] W. Kiesling. Foundations of preferences in database systems. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02)*, pages 311–322, 2002.

[13] J. Lang. A preference-based interpretation of other agents' actions. In S. Zilberstein, J. Koehler, and S. Koenig, editors, *ICAPS*, pages 33–42. AAAI, 2004.

[14] G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: The automated travel assistant. In *Proceedings of the Fifth Internation Conference on User Modeling (UM'97)*, 1997.

[15] K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. Experiments in dynamic critiquing. In R. S. Amant, J. Riedl, and A. Jameson, editors, *IUI*, pages 175–182. ACM, 2005.

[16] D. McSherry. Diversity-conscious retrieval. In *Proceedings of 6th European Conference on Advances in Case-Based Reasoning (ECCBR'02)*, pages 219–233, 2002.

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

[18] R. Price and P. R. Messinger. Optimal recommendation sets: Covering uncertainty over user preferences. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI'05)*, pages 541–548, 2005.

[19] P. Pu and L. Chen. Integrating tradeoff support in product search tools for e-commerce sites. In *Proceedings of ACM Conference on Electronic Commerce (EC'05)*, pages 269–278, 2005.

[20] P. Pu and B. Faltings. Enriching buyers' experiences: the smartclient approach. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'00)*, pages 289–296. ACM Press New York, NY, USA, 2000.

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

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

[23] P. Pu, P. Viappiani, and B. Faltings. Increasing user decision accuracy using suggestions. In *ACM Conference on Human factors in computing systems (CHI06)*, pages 121–130, Montreal, Canada, April 2006.

[24] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Dynamic critiquing. In *Proceedings of the 7th European Conference on Advances in Case-Based Reasoning (ECCBR'04)*, pages 763–777, 2004.

[25] S. Shearin and H. Lieberman. Intelligent profiling by example. In *Proceedings of Intelligent User Interfaces (IUI 2001)*, pages 145–151, 2001.

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

[27] B. Smyth and P. McClave. Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR'01)*, pages 347–361, 2001.

[28] B. Smyth and L. McGinty. The power of suggestion. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico*, pages 127–132, 2003.

[29] M. Stolze and M. Ströbel. Utility-based decision tree optimization: A framework for adaptive interviewing. In M. Bauer, P. J. Gmytrasiewicz, and J. Vassileva, editors, *User Modeling*, volume 2109 of *Lecture Notes in Computer Science*, pages 105–116. Springer, 2001.

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

[31] P. Viappiani, B. Faltings, and P. Pu. Evaluating preference-based search tools: a tale of two approaches. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, pages 205–211, Boston, MA, USA, July 2006. AAAI press.

[32] 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.

[33] P. Viappiani, B. Faltings, and P. Pu. Preference-based search using example-critiquing with suggestions. *Journal of Artificial Intelligence Research (JAIR)*, 27:465–503, 2006.