International Journal of Electronic Commerce

Routledge
Taylor & Francis Group

ISSN: 1086-4415 (Print) 1557-9301 (Online) Journal homepage: https://www.tandfonline.com/loi/mjec20

Adaptive Selection: An Analysis of Critiquing and Preference-Based Feedback in Conversational Recommender Systems

Lorraine McGinty & Barry Smyth

To cite this article: Lorraine McGinty & Barry Smyth (2006) Adaptive Selection: An Analysis of Critiquing and Preference-Based Feedback in Conversational Recommender Systems, International Journal of Electronic Commerce, 11:2, 35-57, DOI: 10.2753/JEC1086-4415110202

To link to this article: https://doi.org/10.2753/JEC1086-4415110202

Published online: 08 Dec 2014.

Adaptive Selection: An Analysis of Critiquing and Preference-Based Feedback in Conversational Recommender Systems

Lorraine McGinty and Barry Smyth

ABSTRACT: E-commerce recommender systems help consumers to locate products within a complex product-space. Conversational recommender systems engage the user in a multi-cycle session, suggesting one or more products during each cycle, and using the feedback to inform the suggestions for the next cycle. By combining user feedback over several cycles, the system obtains a clear picture of the product the user wishes to purchase. As demonstrated under several experimental conditions, the performance of recommender systems is dramatically improved by the technique of adaptive selection, which employs critiquing and preference-based feedback, and emphasizes product diversity rather than similarity as a selection constraint.

KEY WORDS AND PHRASES: Adaptive e-commerce applications, conversational recommender systems, critiquing, feedback elicitation, preference-based feedback, similarity and diversity.

In an e-commerce setting, recommender systems can help to locate suitable products within a complex product-space even when the user has only very limited understanding of product features or of the product-space itself [1, 2, 4–6, 29, 30]. This paper focuses on conversational recommender systems, which facilitate navigation through a product-space by engaging the user in an extended session consisting of a number of recommendation cycles [3, 11, 13–25, 32]. One or more products can be suggested to the user during each cycle, with the feedback informing the suggestions for the next cycle. By combining user feedback over a number of cycles, the recommender system builds a comprehensive representation of the user's product preferences and requirements. Recent research has focused on the characteristics of different forms of user feedback (e.g., suitability to different recommendation scenarios, ability to provide detailed information about a user's preferences, the burden that providing feedback places on the user) in relation to recommendation performance [4–9, 13–20, 32, 34]. Ideally a recommender should guide the user to the ideal product in as short a session as possible and without an overburdening need to provide detailed preference information.

This paper focuses on critiquing and preference-based feedback, both of which are appropriate in many product-recommendation scenarios. *Critiquing* is by far the more common of these two forms of feedback [4–6, 10, 13–15, 18, 22–28, 31]. It allows the user to provide feedback at the level of an individual product feature. For example, "Show me more like product A but cheaper" is a critique on the price feature of a suggested product. Feedback of this type allows the recommender system to limit the set of products for recommendation during the next cycle to products similar to product A but lower-priced. Critiquing is popular because it is informative enough to efficiently guide

1086-4415/2007 $9.50 + 0.00.
DOI 10.2753/JEC1086-4415110202

the recommender system through a complex product-space but does not overburden the user when it comes to providing feedback. *Preference-based feedback* is a far simpler type of feedback, allowing the user to indicate a simple preference for one product suggestion over another— for example, "Show me more items like product B" [6, 7, 16, 17, 25, 26, 32, 34]. One of the benefits of this form of feedback is that it can often be used in situations where the user has very limited understanding of the product features but nonetheless can indicate a product preference (perhaps aided by visual comparison [7]), but the downside is that it offers the recommender system very limited information with which to inform the next cycle. In fact, preference-based feedback is often avoided because it usually leads to protracted recommendation sessions as a result of the limited product information captured as part of user feedback [16, 17, 32, 34].

This paper will show how it is possible to dramatically improve the performance of recommender systems employing these forms of feedback by using a technique called *adaptive selection* [18, 19, 34]. Adaptive selection is distinguished by the way products are selected for recommendation during each recommendation cycle. Recommender systems usually emphasize product similarity as the primary selection constraint—during each cycle the recommender will seek to select the products that are most similar to the user's query[1] or to the current representation of what the user is looking for based on previous cycles. One of the problems with this pure similarity-based approach to recommendation is that it can lead to redundant product suggestions. That is, a set of products may be suggested to the user in a given cycle that are similar not only to the current query but also to each other. If one of these products is unsuitable, then there is a good chance that the alternative suggestions will be equally unsatisfactory. This so-called diversity problem has been highlighted by recent research in the case-based reasoning (CBR)[2] [12] community, and a number of coping strategies have been proposed for improving recommendation diversity in a way that does not compromise similarity to the current query [e.g., 19, 21, 32, 33]. Some relevant related work has been done on the provision of recommendation diversity with the particular aim of repairing users' preference models to make them more complete or to cover their preference uncertainty [10, 23].

Adaptive selection also seeks to introduce diversity into the recommendation process but does so in a unique way, by tuning the degree of diversity and similarity during product selection in response to user feedback (a related approach is treated in [28]). Adaptive selection attempts to mirror how real-life sales personnel adapt their recommendations in response to user feedback by balancing the importance of similarity and diversity. Sometimes a salesperson will make diverse suggestions in an attempt to identify the user's broad interests; other times the salesperson will make less diverse, more similar suggestions in order to focus in on some narrow set of products that appear to be of interest to the user.

Some elements of the ideas discussed in this paper have been presented in previous publications where the original adaptive-selection work focused on improving the efficiency of preference-based conversational recommenders under limited experimental conditions (e.g., datasets) and page constraints

[18, 19, 34]. The main contribution of this paper is its more extensive and comprehensive evaluation of the benefits of adaptive selection as used in both critiquing and preference-based recommenders. Previously unpublished results show how adaptive selection can lead to very significant performance improvements across both types of feedback

Background

This paper is concerned with the general problem of how to guide a recommender system through a complex product-space by using different forms of feedback (in this case, critiquing and preference-based feedback) and a unique approach to item selection that combines similarity and diversity during conversational recommendation. By way of background, the discussion that follows will briefly survey recent research on feedback, similarity, and diversity in recommender systems.

Feedback in Recommender Systems

User feedback is a vital component of most recommenders, allowing them to adapt precisely to the needs of target users, and setting recommender systems apart from more traditional information-retrieval systems. To date researchers have focused on several basic forms of feedback. In addition to critiquing and preference-based forms of feedback, already introduced above, some recommender systems employ value elicitation and ratings-based feedback. *Value elicitation*, a form of feature-based feedback, is similar to critiquing except that users are expected to provide actual feature values as part of their feedback (e.g., "Show me products costing $1,000"), rather than directional preferences. In contrast, *ratings-based feedback* is a form of case-level (or product-level) feedback, like preference-based feedback, except that users are expected to *rate* suggestions instead of just picking one recommendation as their preference.

These feedback strategies can be usefully categorized in terms of the *cost* to the user of providing the feedback, the level of *ambiguity* inherent in the feedback, the level of domain *expertise* required by the user to provide the feedback, and the type of user *interface* needed to capture the feedback (*see Figure 1*).

For example, value elicitation (e.g., "I want a 1-GHz Pentium PC"), perhaps the most common form of feedback, is a rich source of information. Knowing that the user is interested in items with a particular feature allows the recommender to eliminate many irrelevant items from consideration. This form of feedback is often referred to as the "navigation by asking" approach [32] in the context of conversational systems in order to distinguish it from the common initial query specification that often takes place at the start of a recommendation session. Shimazu argues that to provide this level of feedback, the user needs a high level of domain expertise to specify a reasonable feature value [32]. In addition, the user must be willing to answer the direct and specialized questions posed by the recommender, usually selected on the

Feedback	Cost	Ambiguity	Expertise	Interface
Value elicitation	***	*	***	***
Ratings	**	***	**	*
Critique	**	**	**	*
Preference	*	***	*	*

Figure 1. A Comparison of Feedback Strategies

* low; ** moderate; *** high.

basis of their potential information gain. Finally, providing detailed feature-level feedback demands a sophisticated user interface, thus limiting its use on mobile devices.

In contrast, preference-based and ratings-based feedback are low-cost forms of feedback that can be provided by users through a simple interface and with only a rudimentary understanding of the domain. As pointed out in the papers by McGinty and Smyth and by Costello, Doody, McGinty, and Smyth, preference-based feedback is sometimes the only form that is available in certain domains where users are not in a position to provide detailed feature-level feedback but can compare presented items and express a preference at the item (i.e., overall case) level [7, 16, 17, 34]. However, on its own, a simple preference for an item is inherently ambiguous with respect to the user's intent and so has only a limited capacity to guide the recommendation process. Thus, critiquing is often chosen as a useful compromise between value elicitation and preference-based feedback. Instead of providing a specific feature value, the user indicates a feature critique that constrains the value range of the feature in terms of a preferred product. The benefits of critiquing are compelling. It provides a relatively unambiguous indication of the user's current requirement. It is low-cost, easy to implement with the simplest of interfaces, and can be applied even by users with only a moderate understanding of the product-space. Shimazu's ExpertClerk system demonstrates how a combination of straightforward value elicitation and critiquing-based feedback can be more effective than either of the stand-alone approaches [32].

The decision about which form of feedback to choose when developing a given recommender system will depend critically on various characteristics of the recommendation task and domain, and of the likely users. In many consumer applications one cannot assume that users will be experts in a particular product domain, thus eliminating value elicitation as the primary source of feedback. If very little domain expertise can be assumed, then preference-based feedback is the natural choice, whereas confidence in the ability of users to have some level of expertise suggests critiquing. The authors' work has focused mainly on these two forms of feedback.

Regardless of which form of feedback is chosen, special attention must be taken to ensure that it is used to deliver the most efficient recommendation sessions possible—recommendation sessions capable of guiding the user to a satisfactory product in as few cycles as possible. The adaptive-selection approach to conversational recommendation, described in the discussion that follows, is a general platform for delivering efficient recommendation sessions by using preference-based or critiquing forms of feedback. Since the role of diversity in the recommendation process is a critical feature of adaptive selection, the description of the core technique will be preceded by a review of recent research on diversity and similarity in recommender systems.

Similarity Versus Diversity in Recommender Systems

As mentioned above, conversational recommender systems have traditionally followed a similarity-based recommendation policy irrespective of their basic recommendation approaches or the type of feedback they rely on. Accordingly, k cases are selected for recommendation because they are maximally similar to the current user query or the current representation of the user's likely interests—in McSherry's terminology this set of k cases constitutes the *standard retrieval set*, or SRS [20]. Recently, however, there has been a broad acceptance that similarity on its own may not be an ideal selection constraint, especially in recommendation scenarios where only partial information about a user's needs and preferences may be known. Smyth and McClave highlighted the tendency of pure similarity-based approaches to produce recommendations that lack diversity and emphasized the problems that such approaches entailed [33]. In particular, while the recommended product cases may all be similar to the current query, their *value* as a set of alternatives for the user to judge will be compromised if they are also very similar to each other. This observation motivated the need for new selection methods capable of delivering recommendations that are diverse as well as similar to the user query.

The *bounded-greedy* technique introduced by Smyth and McClave was the first attempt to explicitly enhance the diversity of a set of recommendations without significantly compromising their query similarity characteristics [33]. (It is worth nothing that some loss of similarity is experienced with this approach.) A first pass over the recommendable items ranks the available product cases according to their similarity to the current user query. A second pass sequentially transfers cases from this ranked list to the final recommendation list such that at each transfer point the case selected is the one that maximizes the product of its similarity to the target query and its diversity relative to the cases already selected. The diversity of a case c relative to a set of cases C is given by Equation 1. Note that the case most similar to the target query is always transferred first to the final recommendation list.

$$\mathrm{Re}\,lDiv\left(c,C\right)=\frac{\sum_{i=1}^{n}\left(1-Sim\left(c,C_i\right)\right)}{n} \tag{1}$$

In parallel, Shimazu introduced an alternative method for enhancing the similarity of a set of recommendations [32]. In brief, a set of three recommendations, *c1*, *c2*, and *c3*, is chosen relative to some query *q* such that *c1* is maximally similar to *q*, *c2* is maximally dissimilar to *c1*, and then *c3* is maximally dissimilar to *c1* and *c2*. In this way, the triple of cases are chosen to be maximally diverse, but unlike the bounded-greedy technique, the similarity of *c2* and *c3* to the target query is likely to be compromised. The value of this approach is limited to situations where the set of recommended cases is drawn from a set of cases that are all sufficiently similar to the user query to begin with.

A number of alternative diversity-enhancing selection techniques have been proposed. For example, McSherry shows that it is sometimes possible to enhance diversity without loss of query similarity [20]. An approach to enhancing diversity based on the idea of *similarity layers* is described. Very briefly, a set of cases, ranked by their similarity to the target query, can be partitioned into similarity layers, such that all cases in a given layer have the same similarity value to the query. To select a set of *k* diverse cases, the lowest-similarity layer that contributes cases to the SRS is identified, and a subset of cases from this layer is selected for inclusion in the final recommended set; all cases in higher similarity layers are automatically included. Cases are selected from the lowest-similarity layer using an optimal diversity-maximizing algorithm. This approach improves diversity and at the same time fully preserves the similarity of cases to the user query. However, the diversity improvements obtained are typically less than those achieved by the bounded-greedy algorithm.

It is also worth noting that a retrieval technique may not be designed to explicitly enhance diversity but nonetheless have a beneficial effect by its very nature. *Order-based retrieval* is a good example of such a technique [3]. It is based on the idea that the relative similarities of cases to a query of *ideal* feature values is one way of ordering a set of cases for recommendation. Very briefly, order-based retrieval constructs an ordering relation from the query provided by the user and applies this relation to the case-base returning the *k* cases at the top of the ordering. The order relation is constructed from the composition of a set of canonical operators for constructing partial orders based on the feature types that make up the user query. The essential point is that an empirical evaluation of order-based retrieval demonstrates that it has an inherent ability to enhance the diversity of a set of retrieval results—that is, the cases at the top of the ordering tend to be more diverse than an equivalent set of cases ranked on their pure similarity to the user query.

All of the above techniques have been shown to improve the diversity of a single set of recommendations while preserving their similarity to the query to a lesser or greater extent. In other words, using these techniques it is possible to increase the diversity of a given recommendation set in a given recommendation cycle. However, there has been no attempt to assess the implications of such diversity-enhancing methods across the multiple recommendation cycles that make up the dialogue of a conversational recommender system. In particular, most of these techniques operate by eliminating certain cases from the recommended set—cases that would otherwise have been selected on the basis of their similarity to the user query and cases that are not diverse relative to others that have been selected. If one of these cases happens to be the ideal target case for the user, or a case that may lead more directly to the ideal target, then the efficiency (as well as the accuracy) of a conversational recommender may be compromised [19]. The significance of the work presented in this paper is that it looks at the selective introduction of diversity on a cycle-by-cycle basis in a conversational recommender system. As will be seen, the core contribution of the adaptive-selection algorithm concerns a technique for deciding *when to introduce diversity* into product selection and *when to focus on similarity.*

Comparison-Based Recommendation

Comparison-based recommendation was originally introduced by McGinty and Smyth to emphasize the role of feedback in case-based recommenders, and to allow for analysis of preference-based feedback in particular [16]. Very simply, comparison-based shopping sites provide users with the opportunity to compare and contrast recommendation options for product/items of interest and indicate their preferences. The early research focus by McGinty and Smyth looked at how query modification methods might be used with preference-based feedback to produce efficient recommendation dialogues where users find it difficult to provide feature-specific feedback in terms of the recommendation case-descriptions that are presented [16, 17]. The discussion here concerns how the basic algorithm can be easily extended to cater for the critiquing mode of feedback.

The Basic Algorithm

Comparison-based recommendation is a *navigation by proposing* type of recommendation process [32]. The basic algorithm has three main steps: (1) new items are *recommended* to the user based on the current query; (2) the user *reviews* the recommendations and indicates a preferred case; (3) the user's feedback is used to *revise* the query for the next recommendation cycle. The recommendation session terminates when the user is either presented with a suitable item or gives up. A simplified version of the comparison-based algorithm is provided in Figure 2. Preference-based feedback is the assumed feedback strategy, so the preference case in each cycle becomes the query for the next cycle.

Extending Comparison-Based Recommendation for Critiquing

Adapting comparison-based recommendation for critiquing is relatively straightforward. For example, in the travel domain, suppose a user is recommended a $2,000, two-week vacation in Venice, in a three-star hotel, and the feedback indicates that the user is looking for something similar but cheaper (a < *$2,000* critique on the price feature). During the next cycle only items whose price is less than $2,000 will be considered for selection. Similarly, a *!= Sweden* critique on the location feature means that only vacations in locations other than Sweden will be considered. Thus, ***ItemRecommend*** becomes a two-step process. In Step 1, all items that fail to satisfy the critique are eliminated. In Step 2, the remaining items are ranked according to their similarity to the updated query, and the top *k* are selected.

```
Q: Query, CB: Casebase, k: recommendation window-size, i_p: preference case,
R: Recommendation Set, f: case feature

define Comparison-Based-Recommend(Q, CB, k)
begin
  do
    R ←ItemRecommend(Q, CB, k)
    i_p ←UserReview(R, CB)
    Q ←QueryRevise(Q, i_p, R)
  until UserAccepts(i_p)
end

define ItemRecommend(Q, CB, k)
begin
  CB' ←sort cases in CB in decreasing order of their sim to Q
  R ←top k cases in CB'
  return R
end

define UserReview(R, CB)
begin
  i_p ←user selects best case from R
  CB ←CB - R
return c_p

define QueryRevise(Q, i_p, R)
begin
  R' ←R - {i_p}
  for each f_i ∈ i_p
    Q ←transfer f_i
  end for
return Q
end
```

Figure 2. The Basic Comparison-Based Recommendation Algorithm

It is assumed, for the purposes of the present discussion, that query revision follows the simple *more like this* strategy in which the current preferred item is used as the new query in the next recommendation cycle. Four basic types of critiques are used: <, >, =, !=. Obviously, < and > can only be applied to numeric features.

Adaptive Selection

The original comparison-based recommendation framework assumed a similarity-based selection procedure when retrieving suggestions for each cycle. In the real world, however, two selection strategies can be observed in the dialogues that take place between customers and sales personnel. With customers unsure of their exact needs, the salesperson will tend to present a diverse range of alternatives, based on a preliminary set of requirements, to *focus* in on a promising region of a product space. A good salesperson recognizes when customers see something they genuinely like and uses this as a cue to switch the selling strategy to one that tries to *refine* subsequent recommendations in the region of the preference by selecting similar items.

```
Q: Query, CB: Casebase, k: recommendation window-size, i_{p-1}: previous
preference, i_p: preference case, R: Recommendation Set
-----------------------------------------------------------------------
define Comparison-Based-Recommend(Q, CB, k)
   i_{p-1}, i_p ←null
   do
      R ←ItemRecommend(Q, CB, k, i_p, i_{p-1})
      i_{p-1} = i_p
      i_p ←UserReview(R, CB)
      Q ←QueryRevise(Q, i_p, R)
   until UserAccepts(i_p)

define ItemRecommend(Q, CB, k, i_p, i_{p-1})
   if(i_p != null) && (i_p == i_{p-1})
     R ←ReFocus(Q, CB, k)
   else
     R ←ReFine(Q, CB, k)
   return R

define QueryRevise(Q, i_p, R)
    Q ←i_p
    return Q

define UserReview(R, CB)
   ip ←user selects preference item from R
   R ←R - {ip}
   CB ←CB - R
   return ip

define Refine(Q, CB, k)
   CB ←sort CB in decreasing order of their sim to Q
   R ←top k items in CB'
   return R

define Refocus(Q, CB, k)
   return BoundedGreedySelection(Q, CB, k, b)

define BoundedGreedySelection (Q, CB, k, b)
   CB' := bk items in CB that are most similar to Q
   R := {}
   for j := 1 to k
      sort CB' by Quality(Q,i,R) for each case i in CB'
      R  := R + First(CB')
      CB' := CB' - First(CB')
   endFor
   return R
```

Figure 3. The Comparison-Based Recommendation Algorithm with Adaptive Selection Assuming Preference-Based Feedback

The adaptive case-selection technique presented here is motivated by the above observation that different selection strategies are appropriate at different times in the recommendation process. The discussion in this section describes the basic operation of the approach. Figures 3 and 4 illustrate how the basic comparison-based recommendation algorithm is revised accordingly for comparison-based recommenders using preference-based feedback and critiquing.

```
Notation as for Figure 3 as well as: <i_p,c>: user feature critique c over
preference item i_p.

define Comparison-Based-Recommend(Q, CB, k)
   i_p-1, i_p ←null
   do
      R ←ItemRecommend(Q, CB, c, k, i_p, i_p-1)
      i_p-1 = i_p
      <i_p,c> ←UserReview(R, CB)
      Q ←QueryRevise(Q, i_p, R)
   until UserAccepts(i_p)

define ItemRecommend(Q, CB, k, i_p, i_p-1)
   CB' ←{i ∈CB | Satisfies(i,c)}
   if(i_p != null) && (i_p == i_p-1)
      R ←ReFocus(Q, CB, k)
   else
      R ←ReFine(Q, CB, k)
  return R

define QueryRevise(Q, i_p, R)
   Q ←i_p
  return Q

define UserReview(R, CB)
   ip ←user's preferred case from R
   c ←user critique for some f ∈ip
   R ←R - {ip}
   CB ←CB - R
  return <ip,c>

define Refine(Q, CB, k)
   CB ←sort CB in decreasing order of their sim to Q
   R ←top k items in CB'
  return R

define Refocus(Q, CB, k)
  return BoundedGreedySelection(Q, CB, k, b)

define BoundedGreedySelection (Q, CB, k, b)
   CB' := bk items in CB that are most similar to Q
   R := {}
   for j := 1 to k
      sort CB' by Quality(Q,i,R) for each case i in CB'
      R  := R + First(CB')
      CB' := CB' - First(CB')
   endFor
  return R
```

Figure 4. The Comparison-Based Recommendation Algorithm with Adaptive Selection Assuming Critiquing-Based Feedback

Preference Carrying

Unfortunately, it is not immediately obvious how to judge whether the recommender is correctly focused since the target item is unknown and user requirements are typically vague. Adaptive selection solves this by evaluating whether the recommendations made in the *i*th cycle are an improvement on those in the $i - 1$th cycle before choosing a selection strategy for the $i + 1$th cycle. This is achieved by *carrying the preference* item from the previous cycle to the current cycle. Reselection of the carried preference indicates that the user is less satisfied by the $k - 1$ new alternatives presented in that cycle. When this happens (and new items fail to improve on recommendations made in the previous cycle), it means that the recommender is not correctly focused. If the user ignores the carried preference and selects one of the newly recommended items, then the recommender focus of the recommender need not be reset. Carrying the preference item, and monitoring whether or not the user reselects it, enables the implementation of a switching mechanism between two alternative selection strategies: a *refine* strategy that emphasizes similarity and a *refocus* strategy that balances similarity and diversity for improved recommendation coverage.

In theory, carrying the preference limits recommendation coverage because the carried preference takes up a valuable slot in each cycle where a new (i.e., previously unseen) recommendation could have been presented to the user. However, this is easily compensated for because carrying the preference helps protect against *false leads.* That is, if none of the $k - 1$ new cases are relevant, then by reselecting the carried preference, the user is at least maintaining the previous best recommendation rather than being forced to accept a lower-quality recommendation. In practice, this offers, on its own, a substantial improvement to recommendation efficiency.

Refine and Refocus

As mentioned above, the refine strategy makes use of a standard similarity-based selection method, picking $k - 1$ new items that are maximally similar to the current query. The refocus strategy uses the bounded-greedy diversity technique proposed by Smyth and McClave [33] (*see Figures 3 and 4*).

Very briefly, the bounded-greedy technique involves two basic phases. First, the *bk* most similar items to the query are selected (where *b* is typically an integer between 2 and 5). During the second phase, the set (*R*) of selected items is built incrementally. During each step of this build, the remainder of the *bk* items are ordered according to their *quality*, and the highest-quality item is added to *R*. The number of items added to *R* is defined by the window-size parameter, *k,* and this can vary in accordance with such characteristics as screen availability and domain complexity. In the experiments this parameter was set to be 3. The quality of an item *i* is proportional to the similarity between *i* and the current query *q,* and to the diversity of *i* relative to the items so far selected, $\{R = r_1, \ldots, r_m\}$; see Equations (2) and (3).

$$Quality(q, i, R) = \alpha * Sim(q, i) + (1 - \alpha) * Div(i, R) \tag{2}$$

$$\begin{aligned} Div(i, R) &= 1 \text{ if } R = \{\}; \\ &= \Sigma_{j=1..m}(1 - Sim(i, r_j))/m \text{ otherwise.} \end{aligned} \tag{3}$$

The first case to be selected is always the one most similar to the query. In subsequent iterations, the chosen case has the highest quality relative to the query and diversity and cases selected so far. Note the setting of $b = 3$ and $alpha = 0.5$ to balance similarity and diversity during refocusing.

Evaluation

The success of any conversational recommender depends critically on the efficiency of its dialogues, with shorter dialogues likely to lead to greater success than longer ones [8, 20, 32, 34]. This section evaluates the performance of adaptive selection by focusing on the average number of cycles and of unique items that must be presented to a user in a typical recommendation session.[3] For completeness, the evaluations are conducted over a variety of datasets. The results demonstrate that adaptive selection delivers dramatic reductions in dialogue length for both preference-based and critiquing-based recommenders.

Evaluation Setup

Three different datasets were used to evaluate how the adaptive-selection approach affects conversational recommenders that use either critiquing or preference-based feedback:

- A case-base of 585 unique Scotch whiskey cases, where each case is described in terms of 11 unique features, such as *distillery, age, proof, sweetness, peatiness,* and *availability*. A sample case is shown in Figure 5.
- A case-base of 1,024 travel cases, where each case describes a particular vacation in terms of 12 unique features, such as *location, duration, price,* and *number of persons*. A sample case is shown in Figure 6.
- A case-base of 120 PC cases, each describing a unique PC in terms of nine features such as *manufacturer, processor type, processor speed, memory,* and *price*. A sample case is shown in Figure 7.

All three datasets contained both nominal and numeric features, and were subject to the same evaluation methodology. For each case-base, four comparison-based recommenders were compared (two that use preference-based feedback and two that use critiquing).

1. PBF uses the standard preference-based feedback strategy.
2. PBF+AS implements adaptive selection with preference-based feedback.
3. CBF uses the standard critiquing-based feedback strategy.
4. CBF+AS implements adaptive selection with critiquing-based feedback.

CASE ID	DISTILLERY	AGE	PROOF	SWEETNESS	PEATINESS	COLOR	NOSE	FLAVOR/PALATTE	FINISH
Case 500	The Glenlivet	13	40	7	4	gold	sweet	medium-peat	full-body

Figure 5. A Sample Whiskey Case from the Scottish Whiskey Case-Base

CASE ID	REGION	SEASON	TYPE	ACCOMMODATION	DURATION	TRANSPORT	NO. OF PEOPLE	PRICE
Case 1021	Italy	December	Sking	2 star	7 days	plane	2	$ 2500

Figure 6. A Sample Holiday Case from the Travel Case-Base

CASE ID	MANUFACTURER	TYPE	MONITOR SIZE	MEMORY	PROCESSOR	PROCESSOR (MHz)	PRICE
Case 500	DELL	laptop	15'	512 MB	PENTIUM III	900	$ 3500

Figure 7. A Sample PC Case from the PC Case-Base

Evaluation Methodology

Using a leave-one-out methodology, each *case* of a dataset is temporarily removed and used in two ways. First it serves as a *base* case for a set of queries constructed by taking random subsets of item features. Second, the item most similar to the original base is selected. This item serves as the recommendation *target* for the experiments. Thus, the base represents the ideal query for a user, the generated query is the initial query that the user provides to the recommender, and the target is the best available item for the user based on the ideal. During each recommendation cycle, k cases ($k = 3$) are recommended to the user, and the recommendation most similar to the target is chosen as the user's preference. This is the case critiqued (in line with a random feature of the base) in the CBF recommenders (e.g., if the preference case has a Price feature value of $2,000 and the base Price value is *$2,500* a critique of the form [Price, >, $2,000] is used to identify relevant cases for the next cycle). Each query is satisfied when the target case is returned in a recommendation cycle. Finally, for each dataset, three different groups of queries are generated of varying degrees of difficulty (i.e., easy, moderate, difficult)— difficulty is based on the number of cycles required by a standard preference-based recommender.

Recommendation Efficiency

Basic recommendation efficiency can be measured in terms of the average number of cycles (or unique recommendation cases) users must work through before being presented with their ideal target. For each of the three available datasets, the leave-one-out method outlined above is used by all four recommenders to solve all queries in order to measure the mean number of cycles and unique recommendations made to the user.

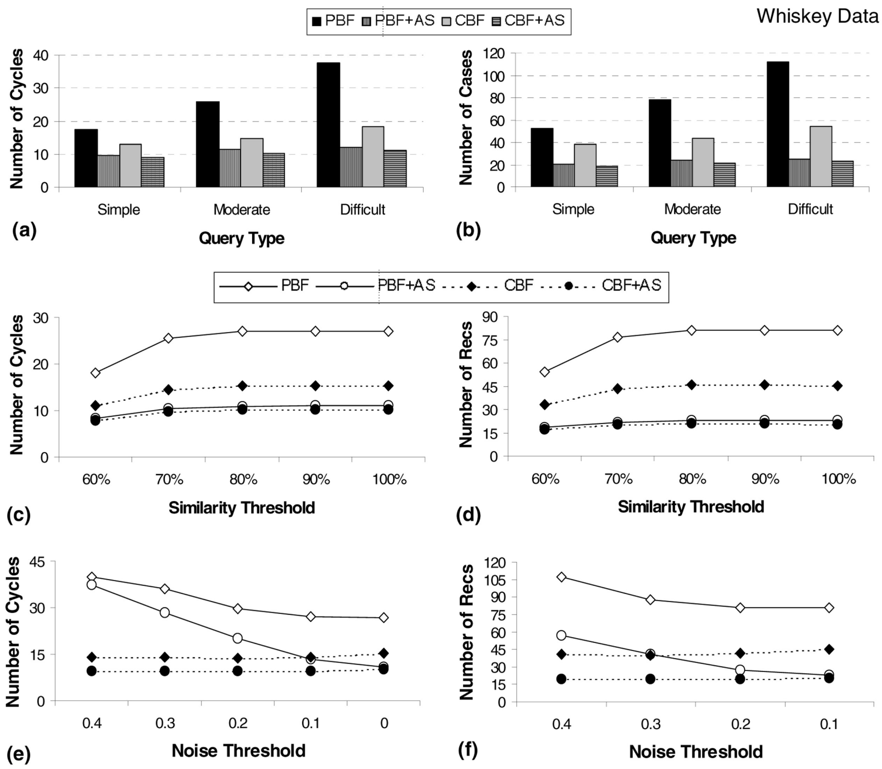

Figure 8. Evaluation Results for the Whiskey Dataset Demonstrating the Benefits That Adaptive Selection Offers to Preference-Based Recommenders and Critiquing-Based Recommenders

Notes: Recommendation efficiency is measured in terms of average number of recommendation cycles, and recommendation items are shown in (a) & (b). Preference tolerance results (i.e., effects of the user selecting a case other than the ideal target case) are shown in graphs (c) and (d). Preference noise results (i.e., the effects of the user selecting a case other than that which is most similar to the target in each cycle) are shown in graphs (e) and (f).

The results for the three datasets are shown in parts (a) and (b) of Figures 8–10 as histograms of mean cycles and recommendation items for each algorithm and query group. The benefits of adaptive selection are clear for both measures, across all datasets, for both feedback approaches. For example, looking at the whiskey data (*see Figure 8*), one sees that PBF requires, on average, 17.5 cycles (and 52.5 unique case retrievals) to satisfy the so-called simple queries, but PBF+AS needs only 9.5 cycles (and 20 case retrievals), a relative reduction of 45 percent in terms of cycles and 62 percent in terms of cases. For the same dataset and query set, CBF requires, on average, 13 cycles (and 38 unique cases), whereas CBF+AS sees a relative reduction in terms of cycles of nearly 31 percent (and 51 percent in terms of the number of unique cases that need be examined and evaluated by the user).

The corresponding results for travel and PC are summarized in (a) and (b) of Figures 9–10 as charts of mean cycles and unique cases for each algorithm

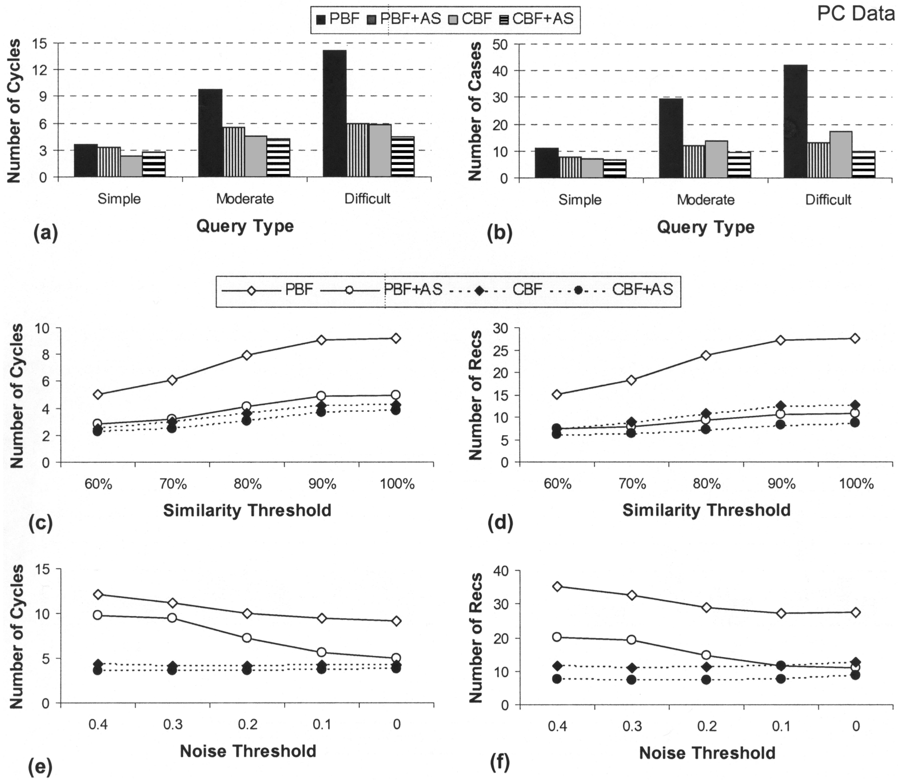

Figure 9. Evaluation Results for the PC Dataset Demonstrating the Benefits That Adaptive Selection Offers to Preference-Based Recommenders and Critiquing-Based Recommenders

Notes: Recommendation efficiency is measured in terms of average number of recommendation cycles, and recommendation items are shown in (a) and (b). Preference tolerance results (i.e., effects of the user selecting a case other than the ideal target case) are shown in graphs (c) and (d). Preference noise results (i.e., the effects of the user selecting a case other than that which is most similar to the target in each cycle) are shown in graphs (e) and (f).

and query group. In travel, for simple queries it takes CBF an average of 9.52 recommendation cycles (and 28.55 unique items) to locate the target items. In contrast, CBF+AS takes 6.7 cycles (and 14 cases), demonstrating once again a relative reduction of 30 percent in cycles and 50 percent in items, compared to standard CBF. Due to the reduced complexity of the PC domain (i.e., fewer features and fewer cases) the benefits across simple queries are less clear than in travel for critiquing, with minor improvements in the number of unique items presented to the user (8 percent CBF+AS over CBF), but minor increases in the number of cycles needed. PBF, however, enjoys enormous reductions of up to 69 percent (for difficult queries) in terms of the number of recommendations (i.e., cases) examined.

Interestingly, the benefits of AS become more pronounced across all three datasets as the level of query difficulty increases. The recommendation dialogues associated with more difficult queries offer adaptive selection a greater

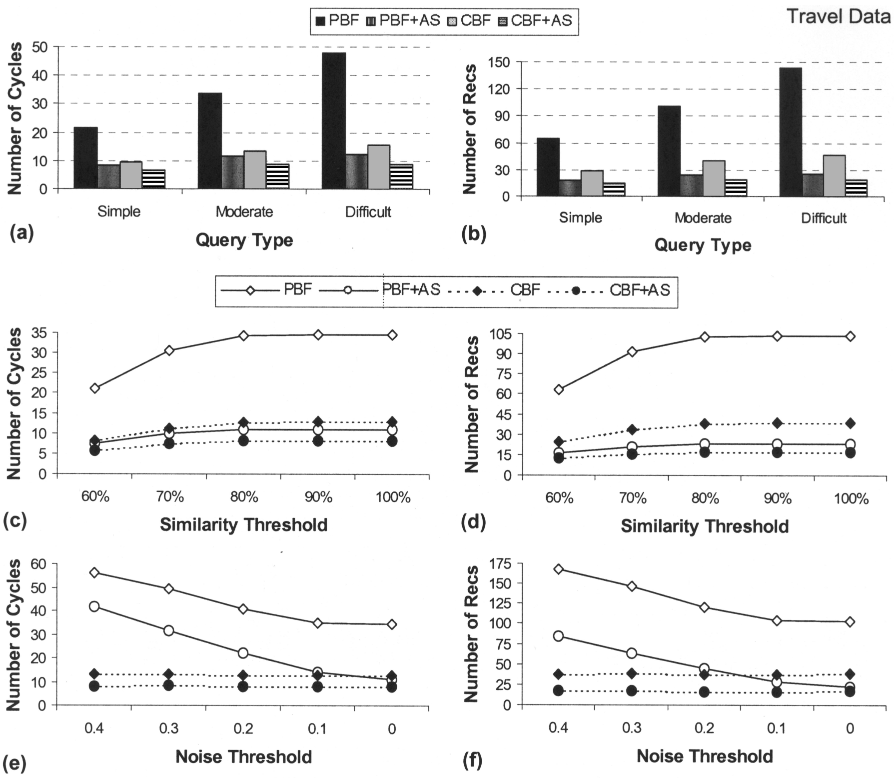

Figure 10. Evaluation Results for the Travel Dataset Demonstrating the Benefits That Adaptive Selection Offers to Preference-Based Recommenders and Critiquing-Based Recommenders

Notes: Recommendation efficiency is measured in terms of average number of recommendation cycles, and recommendation items are shown in (a) and (b). Preference tolerance results (i.e., effects of the user selecting a case other than the ideal target case) are shown in graphs (c) and (d). Preference noise results (i.e., the effects of the user selecting a case other than that which is most similar to the target in each cycle) are shown in graphs (e) and (f).

opportunity for dialogue reduction than the dialogues for simpler queries. For instance, in the whiskey data, the percentage reduction in cycles for PBF+AS, relative to PBF grows from 45 percent (simple) to 68 percent (difficult). For unique recommendation cases it grows from 62 percent to nearly 80 percent. Put another way, satisfying a difficult query takes PBF more than three times as many cycles (and almost 4.5 times as many items) as PBF+AS. In the travel domain, the number of cycles and items required by CBF increases to 15.53 and 46.59 for the moderate and difficult query groups, respectively, and in the PC domain the corresponding values are 5.81 and 17.44. Once again, the increase in cycles and cases proceeds at a much slower rate for CBF+AS, leading to incremental improvements in their benefits relative to CBF. For the difficult queries a dramatic 60 percent reduction in the number of items (compared to CBF) is achieved for CBF+AS. Similarly, in the PC domain, for the difficult queries, a 44 percent reduction in cases is achieved for CBF+AS.

The results over all the examined datasets clearly show a dramatic benefit for adaptive selection. Indeed, for the preference-based whiskey recommenders, the benefit is so great that PBF+AS is capable of satisfying the most difficult queries far more efficiently than PBF can satisfy even the simplest queries! In fact, it takes PBF almost 1.5 times as many cycles (and more than twice as many items) to solve the simplest queries than it takes PBF+AS to solve the most difficult queries. Another point worth highlighting here is that combining preference-based feedback and adaptive selection produces recommendation dialogues significantly shorter than those produced by standard critiquing-based feedback for the most part.

Target Tolerance

A key assumption of the evaluation methodology is that users are only interested in a single target item during recommendation and that the dialogue only terminates when this item is returned. It is perhaps more realistic to assume that the user will be satisfied by any one of a group of items that are similar to the optimal target. The recommenders are reexamined under this more relaxed termination condition by repeating the basic efficiency experiment, except that the dialogue is terminated once an item has been recommended that is within some predefined similarity of the target. A similarity threshold of 70 percent means that the dialogue terminates when an item at least 70 percent similar to the target has been recommended.

Parts (c) and (d) of Figures 8–10 present the results, graphing the mean number of cycles and recommendations made to the user versus the similarity threshold for all queries. As expected, relaxing the termination condition results in shorter recommendation dialogues for all strategies (i.e., PBF, CBR+AS, CBF, CBF+AS). For example, in the whiskey dataset PBF dialogues reduce from just under 35 cycles at the 100 percent similarity threshold (where the optimal item must be recommended) to just over 20 cycles at 60 percent similarity. PBF+AS dialogues reduce from 11 cycles to just 7.4 cycles across the same similarity range. The number of items required is seen to reduce in a similar manner in all datasets. The relative benefits are less for CBF although still significant. For example, staying with the PC dataset, there are cycle reductions of 36 percent for CBF and nearly 32 percent for CBF+AS. Importantly, for both feedback approaches, positive AS benefits are maintained across all similarity thresholds, and can be seen in Figure 9 (c) and (d). In fact, as the threshold increases, and the termination condition becomes more rigid, there are increasing benefits for PBF+AS in terms of cycles and recommendation items. For example, in the travel dataset, the PBF+AS benefit, in terms of cycles, grows from 59 percent at the 60 percent similarity threshold to just under 68 percent at the 100 percent threshold. Once again, adaptive selection delivers significant improvements in recommendation efficiency across a variety of termination conditions for CBF. In terms of unique items, CBF+AS benefits from reductions of 47 percent at the 60 percent similarity threshold to 55 percent at the 100 percent cutoff.

Preference Noise

A second assumption of the evaluation methodology is that the user is capable of correctly selecting/critiquing the best preference item in each recommendation cycle. The experiment relaxed the assumption that the user always prefers the item most similar to the target so as to test whether benefits are found with suboptimal preferences. The above experiment was repeated except that noise was introduced into the preference selection by perturbing the similarities between each recommended item and the target by some random amount within a set noise limit. A 10 percent noise means that each similarity value can change by up to ±10 percent of its actual value. This will potentially change the internal ordering of recommended items during each cycle, resulting in the selection of a preference that may not be the most similar to the target. This approach mimics the situation where users are likely to make preference mistakes more frequently if there is little difference between the target similarities of the recommended items.

Parts (e) and (f) of Figures 8–10 present the results as graphs showing the mean number of cycles and unique recommendations presented to the user versus the noise limit for all queries. As expected, introducing preference noise has a negative impact on the ability of each recommender to locate the target item. For example in the whiskey data (*see Figure 8 (e) and (f)*), PBF dialogues increase from 26.6 cycles at 0 percent noise to 39.9 cycles at 40 percent noise, and PBF+AS dialogues increase from 10.9 cycles to 37.2 cycles. As one would expect, the number of recommendation items also required increases in a similar manner. Perhaps the most notable feature of these results is the lack of observed sensitivity for both critiquing approaches to different levels of noise. It appears that critiquing offers some protection against this type of noise, compensating for suboptimal preferences. The level of protection probably depends on differences in similarity to the target between the preferred/critiqued item and the other items in the recommendation cycle. In the experiments, the difference in similarity to the target between the closest and farthest items from the target, in a single recommendation cycle, was approximately 25–40 percent of the closest item's similarity to the target. This is a significant difference and no doubt explains the degradation in recommendation efficiency observed when stand-alone preference-based feedback is used (i.e., no directional feature critiques are applied over examples). However, the results indicate that critiquing has the ability to overcome such a potential drop in target similarity for the preferred item, presumably because the filtering of items by the chosen critique helps to maintain the relevance of the remaining items for the next cycle.

However, the benefits of adaptive selection remain across all levels of noise, even though the magnitude of the benefits is seen to fall as noise increases for preference-based feedback. For example in the PC dataset, the PBF+AS benefit, in terms of unique recommendation items, falls from 60 percent at the 0 percent noise level to 43 percent at the 40 percent level. Nevertheless, adaptive selection once again offers significant improvements in recommendation efficiency across all datasets even when users make imperfect preference selections for both feedback approaches.

While the results presented here report results averaged over all queries for reasons of brevity, it is worth noting that qualitatively similar results are found for each breakdown of query type (i.e., simple, moderate, and difficult). In previous work the authors have described how these benefits increase with query difficulty [17, 34]. For example, in the whiskey dataset, in terms of unique recommendation items, at the 40 percent noise level, a 27 percent PBF+AS benefit is observed for simple queries, and a 43 percent PBF+AS benefit for difficult queries. For critiquing, in the PC domain the AS benefits are virtually identical for different levels of noise for simple and moderate queries, and not significantly different even for the difficult queries. For example, in PC, the CBF+AS benefit for simple queries is consistently about 10 percent for all levels of noise, rising to about 36 percent for moderate queries, and 41 percent for the difficult queries. A broadly similar pattern is found in the travel domain.

Conclusions

Feedback plays a critical role in many personalized recommender systems, and different types of feedback strike a different balance with respect to user cost, feedback ambiguity, user expertise, and interface requirements. To date, preference-based feedback has been largely ignored by recommender systems. It is an inherently ambiguous form of feedback with limited ability to efficiently guide the recommendation process. Nevertheless, this form of feedback is potentially useful, and may even be vital, in certain recommendation domains where other forms of feedback cannot be used, perhaps because of limited user expertise or even basic device restrictions. More commonplace in on-line recommenders is the critiquing mode of feedback. However, on its own, this form of feedback has also been shown to result in inefficient recommendation dialogues.

The discussion in this paper has described how preference-based feedback and critiquing can both be made more efficient using the adaptive-selection technique, which modifies its recommendation strategy depending on whether or not the recommender is correctly focused on the right region of the recommendation space. This method, used across a variety of experimental conditions and different datasets, demonstrated dramatic performance improvements for both feedback approaches. The length of recommendation dialogues was reduced by up to nearly 80 percent (in the case of PBF) and 60 percent (for CBF).

Furthermore, as was also demonstrated, the use of adaptive selection with preference-based feedback can lead to a recommender system that is more efficient than standard critiquing-based recommenders, thus overturning the conventional wisdom that views preference-based feedback as too inefficient for practical recommendation scenarios. This result, by itself, has the potential to change the perception of simple preference-based feedback, facilitating its practical use in a variety of recommendation scenarios. Interesting avenues for immediate future research on this topic will focus on (1) the opportunities and challenges presented by ratings-based feedback as a means to inform adaptive selection and (2) the exploration of hybrid recommendation approaches that use adaptive selection not only to integrate similarity and diversity retrieval techniques, but also as a means to *swap* between the most appropriate feedback modes in accordance with user interaction cues.

NOTES

1. The term "query" is used here in a broad sense to represent the recommender's current understanding of the user's needs and preferences. In many scenarios a partial query may be provided up-front by the user, and then the elaborated feedback is captured.

2. In CBR recommender systems, a product or item is normally considered a "case," and the recommendation algorithm searches for similar cases/products in the case base.

3. The number of unique recommendation items (cases) is equivalent to the number of cycles × recommendation window size (i.e., here it is static at 3), with repeated cases only counted once.

REFERENCES

1. Bridge, D. Product recommendation systems: A new direction. In D.W. Aha and I. Watson (eds.), *Workshop on CBR in Electronic Commerce.* Vancouver: Springer, 2001, pp. 1–10.
2. Bridge, D. Towards Conversational Recommender Systems: A Dialogue Grammar Approach. In D.W. Aha (ed.), *Proceedings of the Mixed-Initiative Workshop on CBR at the European Conference on Case-Based Reasoning.* Aberdeen, Scotland: Springer, 2002, pp. 9–21.
3. Bridge, D., and Ferguson, A. Diverse product recommendations using an expressive language for case retrieval. In S. Craw and A. Preece (eds.), *Advances in Case-Based Reasoning (Proceedings of the Sixth European Conference).* Aberdeen, Scotland: Springer, 2002, pp. 43–57.
4. Burke, R. Knowledge-based recommender systems. In A. Kent (ed.), *Encyclopedia of Library and Information Systems.* New York: Marcel Dekker, 2000, pp. 32, 69.
5. Burke, R., Hammond, K., and Young, B. Knowledge-based navigation of complex information spaces. In B. Clancey and D. Weld (eds.), *Proceedings of the Thirteenth National Conference on Artificial Intelligence.* Portland, OR: AAAI Press/MIT Press, 1996, pp. 462–468.
6. Chen, L., and Pu, P. Survey of preference elicitation methods. Technical Report No. IC/200467. Lausanne: Swiss Federal Institute of Technology in Lausanne (EPFL), 2004, pp. 1–23.
7. Costello, E., Doody, J., McGinty, L. and Smyth, B. iCARE: Intelligent customer assistance for recommending eyewear. In C. Paris and C.L. Sidner (eds.), *Proceedings of the 11th International Conference on Intelligent User Interfaces.* New York: ACM Press, 2006, pp. 282–284.
8. Coyle, L. and Cunningham, P. Improving recommendation ranking by learning personal feature weights. In P. Gervas and K.M. Gupta (eds.), *Pro-*

ceedings of the Seventh European Conference on Case-Based Reasoning. Madrid: Springer, 2004, pp. 560–572.
9. Doyle, M., and Cunningham, P. A dynamic approach to reducing dialog in on-line decision guides. In E. Blanzieri and L. Portinale (eds.), *Proceedings of the Fifth European Workshop on Case-Based Reasoning.* Trento, Italy: Springer, 2000, pp. 49–60.
10. Faltings, B.; Pu, P.; Torrens, M., and Viappiani, P. Design example-critiquing interaction. In J. Vanderdonckt, N.J. Nunes, and C. Rich (eds.), *Proceedings of the International Conference on Intelligent User Interfaces.* New York: ACM Press, 2004, pp. 22–29.
11. Goker, M., and Thompson, C. Personalized conversational case-based recommendation. In E. Blanzieri and L. Portinale, (eds.), *Proceedings of the Fifth European Workshop on Case-Based Reasoning.* Trento, Italy: Springer-Verlag, 2000, pp. 99–111.
12. Leake, D. *Case-Based Reasoning: Experiences, Lessons and Future Directions.* Menlo Park, CA: AAAI/MIT Press, 1996.
13. McCarthy, K.; Reilly, J.; McGinty, L.; and Smyth, B. On the dynamic generation of compound critiques in conversational recommender systems. In P. De Bra and N. Wolfgang (eds.), *Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems.* Eindhoven: Springer, 2004, pp. 176–184.
14. McCarthy, K.; Reilly, J.; McGinty, L.; and Smyth, B. Experiments in dynamic critiquing. In J. Riedl, A. Jameson, D. Billsus, and T. Lau (eds.), *Proceedings of the International Conference on Intelligent User Interfaces.* San Diego: ACM Press, 2004, pp. 175–182.
15. McCarthy, K., Reilly, J., McGinty, L. and Smyth, B. An analysis of critique diversity in case-based recommendation. In I. Russell and Z. Markov (eds.), *Proceedings of the Special Track on Case-Based Reasoning at the Seventeenth International Conference FLAIRS Conference.* Menlo Park, CA: AAAI Press, 2005, pp. 123–128.
16. McGinty, L. and Smyth, B. Comparison-based recommendation. In S. Craw and A. Preece (eds.), *Proceedings of the Sixth European Conference on Case-Based Reasoning.* Aberdeen, Scotland: Springer, 2002, pp. 575–589.
17. McGinty, L., and Smyth, B. Deep dialogue vs. casual conversation in recommender systems. In F. Ricci and B. Smyth (eds.), *Proceedings of the Workshop on Personalization in eCommerce at the Second International Conference on Adaptive Hypermedia and Web-Based Systems.* Aberdeen, Scotland: Springer, 2002, pp. 80–89.
18. McGinty, L., and Smyth, B. Tweaking critiquing. In B. Mobasher and S. Anand (eds.), *Proceedings of the Workshop on Intelligent Techniques for Personalization as Part of the Eighteenth International Joint Conference on Artificial Intelligence.* Malaga, Spain: Springer, 2003, pp. 20–28.
19. McGinty, L., and Smyth, B. On the role of diversity in conversational recommender systems. In K.D. Ashley and D.G. Bridge (eds.), *Proceedings of the 5th International Conference on Case-Based Reasoning.* Trondheim, Norway: Springer, 2003, pp. 276–290.
20. McSherry, D. Minimizing dialog length in interactive case-based reasoning. In B. Nebel (ed.), *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence.* Seattle: Morgan-Kauffman, 2001, pp. 993–998.

21. McSherry, D. Diversity-conscious retrieval. In S. Craw and A. Preece (eds.), *Proceedings of the Sixth European Conference on Case-Based Reasoning.* Aberdeen, Scotland: Springer, 2002, pp. 219–233.
22. Nader, M.; Ricci, F.; and Bansal, M. Feature selection methods for conversational recommender systems. In Ralph H. Sprague Jr. (ed.), *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service.* Los Alamitos, CA: IEEE Computer Society Press, 2005, pp. 772–777.
23. Price, B., and Messinger, P.R. Optimal recommendation sets: Covering uncertainty over user preferences. In M.M. Veloso and S. Kambhampati (eds.), *Proceedings of the 20th National Conference on Artificial Intelligence.* Los Alamitos, CA: AAAI Press, 2005, pp. 541–548.
24. Pu, P., and Faltings, B. Decision tradeoff using example critiquing and constraint programming. *Constraints, 9*, 4 (2004), 289–310.
25. Pu, P., and Kumar, P. Evaluating example-based search tools. In J.S. Breese, J. Feigenbaum, and M.I. Seltzer (eds.), *Proceedings of the ACM Conference on Electronic Commerce.* New York: ACM Press, 2004, pp. 208–217.
26. Pu, P.; Faltings, B.; and Torrens, M. Effective interaction principles for online product search environments. In J. Liu and N. Cercone (eds.), *Proceedings of The IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology and Web Intelligence.* Los Alamitos, CA: IEEE Computer Society Press, 2004, pp. 724–727.
27. Reilly, J.; McCarthy, K.; McGinty, L.; and Smyth, B. Dynamic critiquing. In P. Funk and M. Gonzlez-Calero (eds.), *Proceedings of the Seventh European Conference on Case-Based Reasoning*. Madrid: Springer, 2004, pp. 763–777.
28. Ricci, F.; Woeber, K.; and Zins, A. Recommendations by collaborative browsing. In A.J. Fren (ed.), *Information and Communication Technologies in Tourism*. New York: Springer, 2005, pp. 172–182.
29. Schafer, J.B.; Konstan, J.; and Riedl, J. E-commerce recommendation applications. *Data Mining and Knowledge Discovery, 5*, 1–2 (2001) 115–153.
30. Shardanand, U., and Maes, P. Social information filtering: Algorithms for automating "word of mouth." In I.R. Katz, R.L. Mack, L. Marks, M.B. Rosson, and J. Nielsen (eds.), *Proceedings of the Conference on Human Factors in Computing Systems.* New York: ACM Press, 1995, pp. 210–217.
31. Shearin, S., and Lieberman, H. Intelligent profiling by example. In J. Lester (ed.), *Proceedings of the ACM Conference on Intelligent User Interfaces.* New York: ACM Press, 2001, pp. 145–151.
32. Shimazu, H. ExpertClerk: Navigating shoppers' buying process with the combination of asking and proposing. In B. Nebel (ed.), *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence.* Seattle: Morgan-Kauffman, 2001, pp. 1443–1448.
33. Smyth, B., and McClave, P. Similarity vs. diversity. In D.W. Aha and I. Watson (eds.), *Proceedings of the Fourth International Conference on Case-Based Reasoning.* Vancouver: Springer-Verlag, 2001, pp. 347–361.
34. Smyth, B., and McGinty, L. The Power of suggestion. In G. Gottlob and T. Walsh (eds.), *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence.* Seattle: Morgan Kaufmann, 2003, pp. 127–132.

LORRAINE MCGINTY (lorraine.mcginty@ucd.ie) is a lecturer in the School of Computer Science and Informatics, University College Dublin. Her primary research interests include adaptive retail, recommender systems, case-based reasoning, personalization, and e-commerce. With more than 35 peer-reviewed scientific papers in leading international conferences and journals, she has received three best-paper awards. She has served on the program committees for several international conferences, was workshop co-coordinator for the Fifth International Conference on Case-Based Reasoning and general program chair/editor for the Fourteenth National Artificial Intelligence and Cognitive Science Conference, and recently guest edited the *Artificial Intelligence Review Journal*. She is also on the organization committee and is chair of the Doctoral Consortium for the 2006 International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems.

BARRY SMYTH (barry.smyth@ucd.ie) holds the Digital Chair of Computer Science, is an ECCAI Fellow, is head of the School of Computer Science and Informatics, University College Dublin, and was a founder of ChangingWorlds Ltd., a leading developer of personalized portal platforms for the mobile sector. He received a B.Sc. in computer science from University College Dublin and a Ph.D. from Trinity College, Dublin. He works in several areas of artificial intelligence including personalization, case-based reasoning and information retrieval, has authored more than 200 technical papers, and received best-paper awards at the International Joint Conference on Artificial Intelligence and the European Conference on Artificial Intelligence.