

Explanation in Recommender Systems

David McSherry

School of Computing and Information Engineering, University
of Ulster, Coleraine BT52 1SA, Northern Ireland
dmg.mcsberry@ulster.ac.uk

Abstract. There is increasing awareness in recommender systems research of the need to make the recommendation process more transparent to users. In recent work we presented a conversational CBR approach to product recommendation that uniquely combines an effective strategy for reducing the length of recommendation dialogues with a mechanism for ensuring that the dialogue is terminated only when it is certain that the recommendation will be the same no matter how the user chooses to extend her query. Our approach has the advantage of enabling recommendations to be *justified* on the grounds that any remaining attributes cannot affect the solution. As we show in this paper, the relevance of any question the user is asked can also be explained in terms of its ability to discriminate between competing cases, thus giving users a unique insight into the recommendation process. We present an extended version of our conversational CBR approach in a mixed-initiative recommender system called *Top Case* and demonstrate its ability to explain the relevance of any question the user is asked and justify its recommendations.

1 Introduction

The importance of intelligent systems having the ability to explain their reasoning is well recognised in domains such as medical decision making and intelligent tutoring [1-2]. In an intelligent tutoring system, for example, communicating the reasoning process to students may be as important as finding the right solution. Until recently, explanation in recommender systems appears to have been a relatively neglected issue. However, recent research has highlighted the importance of making the recommendation process more transparent to users and the potential role of explanation in achieving this objective [3-5].

Herlocker *et al.* [3] suggest that the “black box” image of recommender systems may be one of the reasons why they have gained much less acceptance in high-risk domains such as holiday packages or investment portfolios than in low-risk domains such as CDs or movies. They argue that extracting meaningful explanations from the computational models on which recommendations are based is a challenge that must be addressed to enable the development of recommender systems that are more understandable, more effective, and more acceptable. It is an argument that seems equally compelling in collaborative and content-based approaches to product recommendation.

In recent work we presented a conversational CBR approach to product recommendation that uniquely combines an effective strategy for reducing the length

of recommendation dialogues with a mechanism for ensuring that the dialogue is terminated only when it is certain that the recommendation will be the same no matter how the user chooses to extend her query [6]. As well as ensuring that there can be no loss of solution quality relative to a full-length query, our approach has the advantage of enabling recommendations to be *justified* on the grounds that any remaining attributes cannot affect the solution. As we show in this paper, the relevance of any question the user is asked can also be explained in terms of its ability to discriminate between competing cases, thus giving users a unique insight into the recommendation process.

In Section 2, we examine existing approaches to explanation in recommender systems and some of the lessons learned from this research. In Section 3, we present our conversational CBR approach to product recommendation as implemented in a mixed-initiative recommender system called *Top Case*. In Section 4, we describe how explanations of the recommendation process are automatically generated in *Top Case* with no requirement for domain knowledge other than the similarity knowledge and cases available to the system at run time. An example recommendation dialogue based on a well-known case library in the travel domain is used to demonstrate how *Top Case* can explain the relevance of any question the user is asked and justify its recommendations. Our conclusions are presented in Section 5.

2 Existing Approaches

Herlocker *et al.* [3] evaluated several explanation interfaces for their collaborative movie recommender *MovieLens* in terms of their effects on user acceptance of the system's recommendations. The most convincing explanation of why a movie was recommended was one in which users were shown a histogram of the ratings of the same movie by similar users. Moreover, grouping together of good ratings (4 or 5) and bad ratings (1 or 2) and separation of ambivalent ratings (3) was found to increase the effectiveness of the histogram approach. Interestingly, the second most convincing explanation was a simple statement of the system's performance in the past e.g.

☺ MovieLens has predicted correctly for you 80% of the time in the past

Another important finding was that some of the explanations evaluated actually had a *negative* impact on acceptance, showing that no explanation may be better than one that is poorly designed.

CBR recommender systems that can explain their recommendations include Shimazu's *ExpertClerk* [5] and our own *First Case* [4]. *ExpertClerk* can explain why it is proposing two contrasting products in terms of the trade-offs between their positive and negative features e.g.

☺ This blouse is more expensive but the material is silk. That one is cheaper but the material is polyester

Its explanations are based on assumed preferences with respect to attributes not mentioned in the user's query. For example, a blouse made of silk is assumed to be preferred to one made of polyester.

In a similar way, First Case can explain why one case is more highly recommended than another by highlighting the benefits it offers [4]. As the following example illustrates, it can also explain why a given product, such as a personal computer, is recommended in terms of the *compromises* it involves with respect to the user's preferences.

- ☺ Case 38 differs from your query only in speed and monitor size. It is better than Case 50 in terms of memory and price

However, the potential role of explanation in recommender systems is not limited to explaining why a particular item is recommended. Later in this paper, we present a CBR recommender system that can explain the relevance of any question the user is asked. In recommender systems that treat some or all of the user's requirements as constraints that must be satisfied, explanation can also play an important role in assisting the process of recovery from the retrieval failures that occur when there is no exact match for the user's requirements [7-8].

Hammond *et al.*'s [7] *Car Navigator* is a recommender system for cars that uses declarative knowledge to explain trade-offs that are known to be common causes of retrieval failure in the domain, such as that between fuel economy and horsepower. For example, if the user asks for good fuel economy *and* high horsepower, she is shown a movie explaining the trade-off between these features. The user is also advised that she will need to revise her preferences if she hopes to find a car that meets her requirements.

In recent research, we combined a *knowledge-light* approach to explanation of retrieval failure with a mixed-initiative approach to recovery from retrieval failure in a CBR recommender system called *ShowMe* [8]. A retrieval failure in ShowMe triggers an explanation of "what went wrong" that draws the user's attention to combinations of features in her query for which there are no matching cases e.g.

- ☺ Sorry, there are no products that match these combinations of features in your query: (price \leq 700, type = laptop), (type = laptop, screen size = 19)

As well as highlighting areas of the product space in which the case library is lacking in coverage, the explanation may reveal *misconceptions* on the part of the user such as the price she expects to pay for the product she is seeking. Showing the user only the minimally failing *sub-queries* of her query, a technique we have adapted from research on "co-operative" responses to failing database queries [9], helps to minimise cognitive load in the approach. Explanation of the retrieval failure is followed in ShowMe by a mixed-initiative recovery process in which the user is guided in the selection of one or more constraints to be eliminated from her query [8].

3 Recommendation in Top Case

We now present our conversational CBR approach to product recommendation as implemented in a mixed-initiative recommender system called *Top Case*. As in other conversational CBR approaches, a query is incrementally elicited in an interactive dialogue with the user. One distinguishing feature of our approach is a *goal-driven*

attribute selection strategy that has been shown to be very effective in reducing the length of recommendation dialogues [6]. Another is a simple mechanism for recognising when the dialogue can be terminated without loss of solution quality.

In the following discussion, we assume that the similarity of any case C to a given query Q over a subset A_Q of the case attributes A is defined to be:

$$Sim(C, Q) = \sum_{a \in A_Q} w_a sim_a(C, Q)$$

where for each $a \in A$, w_a is the importance weight associated with a and $sim_a(C, Q)$ is a local measure of the similarity of $\pi_a(C)$, the value of a in C , to $\pi_a(Q)$, the preferred value of a . We also assume that for each $a \in A$, $0 \leq sim_a(C, Q) \leq 1$ and $sim_a(C, Q) = 1$ if and only if $\pi_a(C) = \pi_a(Q)$.

3.1 Identifying Dominated Cases

A key role in our conversational CBR approach to product recommendation is played by the concept of case dominance that we now define.

Definition 1. A given case C_2 is dominated by another case C_1 with respect to a query Q if $Sim(C_2, Q) < Sim(C_1, Q)$ and $Sim(C_2, Q^*) < Sim(C_1, Q^*)$ for all possible extensions Q^* of Q .

One reason for the importance of case dominance in product recommendation is that if a given case C_2 is dominated by another case C_1 then the product represented by C_2 can be eliminated. It can also be seen that if the case that is most similar to the user's current query dominates all other cases, then there is no need for the query to be further extended as the user's preferences with respect to any remaining attributes cannot affect the recommendation. Of course, the number of ways in which a given query can be extended may be very large. So given an incomplete query Q and cases C_1, C_2 such that $Sim(C_2, Q) < Sim(C_1, Q)$, how can we tell if C_2 is dominated by C_1 without resorting to exhaustive search?

One situation in which C_2 is clearly dominated by C_1 is when both cases have the same values for all the remaining attributes. Another is when $Sim(C_1, Q) - Sim(C_2, Q)$ is greater than the sum of the importance weights of all the remaining attributes. In situations where dominance is less obvious, account must be taken of the similarity between the two cases as well as their similarities to the current query. The criterion used to identify dominated cases in Top Case is stated in the following theorem [6].

Theorem 1. A given case C_2 is dominated by another case C_1 with respect to a query Q if and only if:

$$Sim(C_2, Q) + \sum_{a \in A - A_Q} w_a (1 - sim_a(C_1, C_2)) < Sim(C_1, Q)$$

Our theorem assumes that for each $a \in A$, the distance measure $d_a = 1 - sim_a$ satisfies the triangle inequality. An alternative dominance criterion that does not rely on the triangle inequality is presented in [6].

3.2 Attribute Selection Strategy

An initial query entered by the user is incrementally *extended* in Top Case by asking the user to specify preferred values for attributes not mentioned in her initial query. The attribute selected by Top Case on each cycle of the recommendation process is the one that is most useful for confirming the case it has selected as the *target* case. A target case is initially selected at random from the cases that are maximally similar to the user's initial query, and continually revised as the query is extended. No change is needed as long as the target case remains one of the cases that are maximally similar to the current query.

As Fig. 1 illustrates, attribute selection in Top Case aims to maximise the number of cases dominated by the target case. The cases currently dominated by the target case are shown in the lower half of the diagram. As indicated by the dashed arrows, there may be many dominance relationships with respect to the current query, but Top Case confines its attention to cases dominated by the target case. For each of the remaining attributes, it uses the dominance criterion from Theorem 1 to determine the number of cases that will be dominated by the target case if the preferred value of the attribute is the same as in the target case. It then selects the attribute that maximises the number of cases potentially dominated by the target case. If two or more attributes are equally promising according to this criterion, Top Case uses the importance weights associated with the case attributes as a secondary selection criterion. That is, it chooses the most important of the equally promising attributes.

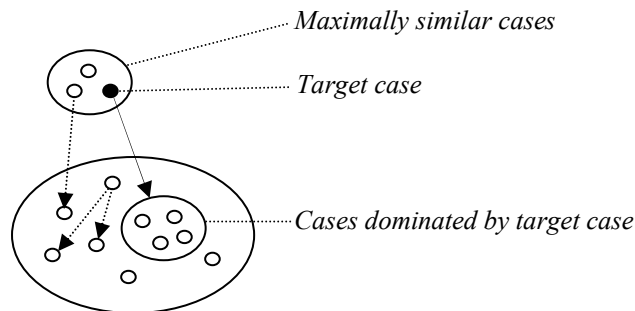


Fig. 1. Attribute selection in Top Case aims to maximise the number of cases dominated by the target case

3.3 Terminating the Recommendation Dialogue

On each cycle of the recommendation process, the user is asked for the preferred value of the most useful attribute and shown the cases that are now most similar to her query. The user can terminate the recommendation dialogue at any stage by selecting one of the cases she is shown as the product she prefers. Otherwise, query elicitation continues until Top Case has determined that its recommendation based on the current query cannot be affected by the user's preferences with respect to any remaining attributes. At this point, the dialogue is terminated and the user is informed

that the target case has been confirmed as the recommended case. Any case that equals the current similarity of the target case at this stage is also recommended.

As we have argued in [6], naïve approaches to terminating recommendation dialogues such as stopping when the similarity of any case reaches a predefined threshold cannot guarantee that a better solution will not be found if the dialogue is allowed to continue. In Top Case, the recommendation dialogue is terminated only when it is *certain* that there can be no loss of solution quality relative to a full-length query representing the preferences of the user with respect to all the case attributes. The criteria used in Top Case to recognise when the recommendation dialogue can be safely terminated are stated in the following theorem, the proof of which follows easily from the theory presented in [6].

Theorem 2. *The target case, and any case that equals its similarity to the current query, can be recommended without loss of solution quality relative to a full-length query if and only if the following conditions hold:*

1. No case is more similar to the current query than the target case
2. Any case that equals the similarity of the target case to the current query has the same values as the target case for all remaining attributes
3. All cases that are less similar than the target case are dominated by the target case

It is worth noting that *no* other criteria for termination of the recommendation dialogue can guarantee that a more similar case will not be found if the dialogue is allowed to continue.

4 Explanation in Top Case

Our conversational CBR approach to product recommendation uniquely combines an effective strategy for reducing the length of recommendation dialogues with a mechanism for ensuring that the dialogue is terminated only when the user's preferences with respect to any remaining attributes cannot affect the solution [6]. An important advantage in comparison with approaches that rely on arbitrary thresholds to decide when the dialogue should be terminated is that recommendations can be *justified* on the grounds that any remaining attributes cannot affect the solution. As we show in this section, the relevance of any question the user is asked can also be explained in terms of the system's strategy of eliminating competing cases and ultimately confirming the target case.

The approach to explanation in Top Case that we now present requires no domain knowledge other than the similarity knowledge and cases available to the system at run time. As well as explaining the relevance of any question it asks the user, Top Case can explain why it is recommending a particular case. Both types of explanation are dynamically generated at run time using simple templates.

4.1 Explanation Templates

The template used by Top Case to explain the relevance of a question the user is asked depends on whether or not the question has the potential to confirm the target

case as the recommended case. Usually in the early stages of query elicitation, the selected attribute can at best be expected to increase the number of cases dominated by the target case rather than confirm the target case outright. The explanation template used when the target case cannot be confirmed in a single step is:

- ☺ Because if $a = v$ this will increase the similarity of Case X from S_1 to S_2
{and eliminate N cases [including Cases X_1, X_2, \dots, X_r]}

where:

- a is the attribute whose preferred value the user is asked to specify
- v is the value of a in the target case
- Case X is the target case
- S_1 is the similarity of the target case to the current query
- S_2 is the similarity of the target case that will result if the preferred value of a is v
- N is the number of cases that will be eliminated if the preferred value of a is v
- Cases X_1, X_2, \dots, X_r are cases that the user was shown on the previous recommendation cycle that will be eliminated if the preferred value of a is v

The section of the template enclosed in curly brackets is used only if one or more cases will be eliminated if the preferred value of a is v , which may not be the case in the early stages of query elicitation. The section enclosed in square brackets is used only if one or more of the cases that the user was shown in the previous recommendation cycle will be eliminated if the preferred value of a is v .

The template used by Top Case to explain the relevance of a question that does have the potential to confirm the target case in a single step is:

- ☺ Because if $a = v$ this will confirm Case X as the recommended case

where a , v , and Case X are as defined for the previous template.

The template used by Top Case to explain why it is recommending a particular case depends on whether that case exactly matches the user's query and whether preferred values have been elicited for all case attributes. In the example template below, Case X is the recommended case, *attributes-1* are attributes in which it differs from the user's query, and *attributes-2* are attributes for which preferred values have not been elicited because they cannot affect the recommendation.

- ☺ Case X differs from your query only in *attributes-1* and is the best case no matter what *attributes-2* you prefer

As in First Case [4], the aim of highlighting any compromises that the recommended case involves is to help the user decide whether or not to accept the system's recommendation. Also informing the user that her preferences with respect to any remaining attributes cannot affect the solution may help to increase her *confidence* in the recommendation. How effective the explanations provided by Top Case are in achieving these objectives is an important issue to be addressed by further research.

4.2 Example Dialogue

The example dialogue in Fig. 2 is based on the *Travel* case library (www.ai-cbr.org), a standard benchmark containing the descriptions of over 1,000 holidays. Attributes in the case library and importance weights assigned to them in Top Case are price (8), month (7), region (6), persons (5), duration (4), type (3), accommodation (2), and transport (1). Minor editing of the example dialogue in the interest of clarity includes the format of the user's initial query. The current similarity of each retrieved case, normalised by the sum of all the importance weights, is shown in brackets.

In response to her initial query, the user is shown Case 510, the case initially selected by Top Case as the target case, and the two most similar of the 972 cases that are not currently dominated by the target case. Having taken the initiative, Top Case now selects region as the most useful attribute for confirming the target case and asks the user what region she prefers.

When asked to explain the relevance of region, Top Case points out that if the preferred region is *tyrol*, this will increase the similarity of the target case from 0.28 to 0.44 and eliminate 866 of the 972 competing cases. When the user chooses *alps* instead as the preferred region, the target case changes to Case 574, but now there are only 82 competing cases. The user's answers to the next two questions are enough for Top Case to confirm Case 574 as the recommended case. When asked to explain its recommendation, Top Case points out that the recommended case differs from the user's query only in price and that her preferences with respect to the remaining attributes cannot affect the recommendation.

4.3 Discussion and Related Work

A known limitation of similarity-based retrieval is that the most similar case may not be the one that is most acceptable to the user [4]. It must therefore be recognised that the case recommended by Top Case may not be acceptable to the user even though it is guaranteed to remain the most similar case no matter how she chooses to extend her query. In future research we plan to investigate an approach to addressing this issue in which the dialogue is allowed to continue beyond the initial recommendation so that the user can extend or revise her query to include one or more constraints that must be satisfied.

It is worth noting that all three cases presented by Top Case in response to the user's initial query have the same values for holiday type and month; in fact there are 19 other cases that exactly match the user's initial query. This is a common problem associated with incomplete queries [10] that highlights the importance of Top Case's ability to take the initiative in helping users discriminate between alternatives that are equally good in terms of their initial requirements.

Allowing the user to enter an initial query to be incrementally extended is a feature that Top Case shares with mixed-initiative CBR tools for fault diagnosis such as NaCoDAE [11]. Most of the dialogue features associated with mixed-initiative interaction in CBR [11-12] are in fact supported by Top Case, though not all are shown in the example dialogue. At any stage of the recommendation dialogue, for example, the user can specify a preferred value for an attribute other than the one considered most useful by Top Case.

Top Case: Please enter your query.

User: type = wandering, month = aug

Top Case: The target case is:
Case 510: 609, aug, tyrol, 5, 7, wandering, flat, car (0.28)
Other competing cases (972) include:
Case 702: 1698, aug, giant_mountains, 3, 14, wandering, star2, car (0.28)
Case 574: 529, aug, alps, 4, 7, wandering, flat, car (0.28)
What is the preferred region?

User: **why**

Top Case: Because if region = tyrol this will increase the similarity of Case 510 from 0.28 to 0.44 and eliminate 866 cases, including Case 574
What is the preferred region?

User: alps

Top Case: The target case is:
Case 574: 529, aug, alps, 4, 7, wandering, flat, car (0.44)
Other competing cases (82) include:
Case 570: 799, aug, alps, 4, 7, wandering, flat, car (0.44)
Case 586: 1958, aug, alps, 6, 14, recreation, flat, car (0.36)
What is the preferred price?

User: **why**

Top Case: Because if price = 529 this will increase the similarity of Case 574 from 0.44 to 0.67 and eliminate 72 cases, including Case 570
What is the preferred price?

User: 500

Top Case: The target case is:
Case 574: 529, aug, alps, 4, 7, wandering, flat, car (0.67)
Other competing cases (10) include:
Case 74: 814, aug, alps, 6, 7, recreation, flat, train (0.57)
Case 586: 1958, aug, alps, 6, 14, recreation, flat, car (0.54)
What is the preferred number of persons?

User: **why**

Top Case: Because if number of persons = 4 this will confirm Case 574 as the recommended case
What is the preferred number of persons?

User: 4

Top Case: The recommended case is Case 574 (0.8)

User: **why**

Top Case: Case 574 differs from your query only in price and is the best case no matter what transport, duration, or accommodation you prefer

Fig. 2. Example recommendation dialogue in Top Case based on the Travel case library

5 Conclusions

There is increasing awareness in recommender systems research of the need to make the recommendation process more transparent to users [3-5]. We have presented a mixed-initiative recommender system that can explain the relevance of any question the user is asked and justify its recommendations on the grounds that any remaining attributes cannot affect the solution. Our system uses a conversational CBR approach to product recommendation that uniquely combines an effective strategy for reducing the length of recommendation dialogues with a mechanism for ensuring that the dialogue is terminated only when there can be no loss of solution quality relative to a full-length query [6]. In future research we plan to investigate the potential impact of the system's explanation capabilities and ability to support mixed-initiative interaction on the effectiveness of the recommendation process.

References

1. Armengol, E., Paludàries, A., Plaza, E.: Individual Prognosis of Diabetes Long-Term Risks: a CBR Approach. *Methods of Information in Medicine* **40** (2001) 46-51
2. Sørmo, F., Aamodt, A.: Knowledge Communication and CBR. In: González-Calero, P. (ed.) *Proceedings of the ECCBR-02 Workshop on Case-Based Reasoning for Education and Training* (2002) 47-59
3. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining Collaborative Filtering Recommendations. *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (2000) 241-250
4. McSherry, D.: Similarity and Compromise. In: Ashley, K.D., Bridge, D.G. (eds.) *Case-Based Reasoning Research and Development*. LNAI, Vol. 2689. Springer-Verlag, Berlin Heidelberg New York (2003) 291-305
5. Shimazu, H.: ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. *Artificial Intelligence Review* **18** (2002) 223-244
6. McSherry, D.: Increasing Dialogue Efficiency in Case-Based Reasoning Without Loss of Solution Quality. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* (2003) 121-126
7. Hammond, K.J., Burke, R., Schmitt, K.: A Case-Based Approach to Knowledge Navigation. In: Leake, D.B. (ed.) *Case-Based Reasoning: Experiences, Lessons & Future Directions*. AAAI Press/MIT Press, Menlo Park, CA (1996) 125-136
8. McSherry, D.: Incremental Relaxation of Unsuccessful Queries. In: González-Calero, P., Funk, P. (eds.) *Advances in Case-Based Reasoning*. LNAI, Vol. -. Springer-Verlag, Berlin Heidelberg New York (2004)
9. Gaasterland, T., Godfrey, P., Minker, J.: An Overview of Cooperative Answering. *Journal of Intelligent Information Systems*, **1** (1992) 123-157
10. McSherry, D.: The Inseparability Problem in Interactive Case-based Reasoning. *Knowledge-Based Systems* **15** (2002) 293-300
11. Aha, D.W., Breslow, L.A., Muñoz-Avila, H.: Conversational Case-Based Reasoning. *Applied Intelligence* **14** (2001) 9-32
12. McSherry, D.: Interactive Case-Based Reasoning in Sequential Diagnosis. *Applied Intelligence* **14** (2001) 65-76