Recommendation Engineering

David McSherry[1]

Abstract. We present a new approach to product recommendation that addresses the limitations of the standard case-based reasoning (CBR) approach of retrieving a list of cases that are most similar to a target query. Instead, the target query is used to construct a retrieval *network* in which cases selected for initial presentation to the user are *representative* of cases that differ from the target query in similar ways. By following links in the retrieval network, the user can examine alternative solutions with no need to await the retrieval of new cases. Other advantages of the approach include increased diversity among the cases initially presented to the user and the ability to explain why cases are recommended in much the same way as a human salesperson might explain their relevance.

1 INTRODUCTION

Recommender systems for helping customers to select products or services are increasingly common in electronic commerce. An important advantage of case-based reasoning (CBR) as an approach to product recommendation is the ability to suggest alternatives that may be acceptable when none of the available solutions exactly matches the user's requirements. However, recent research has highlighted the limitations of the standard CBR approach of retrieving a list of cases that are most similar to a target query.

Ferguson and Bridge [1] argue that customers need to understand why cases have been recommended, and identify *spurious precision* as a complicating factor in the interpretation of retrieval results. In a similar vein, Wilke et al. [2] identify the ability to *explain* the relevance of retrieved cases to the user's query as an important requirement in intelligent sales support systems. Hammond et al.'s [3] insight that users often find it easier to critique a specific example than formulate queries highlights the importance of query *refinement* based on changes suggested by the user [2-5].

There is also growing awareness of the need for recommender systems to offer a more *diverse* set of alternatives than is possible by simply retrieving the cases that are most similar to a target query. The problem is that the most similar cases are also likely to be very similar to each other, with the result that the user may be offered a very limited choice. Smyth and Cotter [6] combine CBR with other recommendation techniques that are less susceptible to the so-called *diversity* problem. More recent research has focused on solutions that remain within the CBR paradigm, for example by combining measures of similarity and diversity in the retrieval process to achieve a better balance between these often conflicting characteristics of the retrieved cases [7-8]. In this paper, we present an algorithm for *recommendation engineering* called *R-Net*. Instead of simply presenting the user with a list of recommended cases, R-Net builds a retrieval *network* in which the cases it selects for initial presentation to the user are *representative* of cases that differ from the target query in similar ways. By following links in the retrieval network, the user can explore the solution space in the neighbourhood of her query with no need to await the retrieval of new cases. Other advantages of the approach include increased diversity among the cases initially presented to the user and the ability to explain why cases are recommended in much the same way as a human salesperson might explain their relevance.

2 THE RETRIEVAL SET

We refer to the set of cases presented to the user by a recommender system as the *retrieval set*. The *standard* retrieval set for a target query Q consists of the k cases that are most similar to Q. In practice, the value of k may be dictated by the available screen size, chosen by system designers in accordance with HCI principles, or configurable by the user [7-9]. Before describing how the retrieval set is constructed in R-Net, we introduce the similarity and diversity measures used in our experiments and the example case library that we use to illustrate the approach.

2.1 Similarity and diversity

Increasing diversity in the retrieval set for a target query often means decreasing the average similarity of the retrieved cases to the target query relative to the standard retrieval set [7-8]. Often in practice, queries are *incomplete* in the sense that preferred values are specified for only some of the case attributes, thus reducing the number of attributes available for retrieval [10]. Given a query Q, we denote by A_Q the set of attributes for which preferred values are specified. We refer to $|A_Q|$ as the *length* of the query. The *matching features* similarity measure used in our experiments has the advantage of being domain independent and is a special case of the standard *weighted-sum* measure.

For any case C, we define:

$$Sim_{MF}(C, Q) = \frac{\left|\{a \in A_Q : \pi_a(C) = \pi_a(Q)\}\right|}{\left|A_Q\right|} \qquad (1)$$

where for all $a \in A_Q$, $\pi_a(C)$ is the value of a for C and $\pi_a(Q)$ is the preferred value of a as specified in Q.

[1] School of Information and Software Engineering, University of Ulster, Coleraine BT52 1SA, Northern Ireland

A point we would like to emphasise is that the proposed approach to product recommendation is not specific to Sim_{MF} and can be used with any similarity measure. While a measure of diversity is not required in recommendation engineering, such a measure is required to evaluate the approach. For this purpose we use the measure proposed by Smyth and McClave [7]. Given a target query Q, similarity measure Sim, and retrieval set $R = \{C_1, C_2, \ldots C_k\}$, the diversity of R is:

$$diversity(R, Q) = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} (1 - Sim(C_i, C_j)) \qquad (2)$$

An important point to note is that we measure similarity, and hence diversity, with respect to the attributes in a given query only.

The algorithm we use as a benchmark in our evaluation of R-Net as a diversification technique is Smyth and McClave's *Bounded Greedy* (BG) algorithm [7]. In BG, cases are sequentially selected for addition to a retrieval set initially containing only the most similar case until the retrieval set contains the required number of cases. The case selected at each stage is the one that maximises the product of its similarity to the target case and its diversity *relative* to the cases that have been selected so far. BG is bounded in the sense that candidates for addition to the retrieval set are restricted to the $2k$ cases that are most similar to the target query, where k is the required size of the retrieval set.

The effectiveness of a diversification technique can be measured in terms of the *relative benefit* it provides, defined as the average increase in diversity relative to the standard retrieval set divided by the average decrease in similarity. BG has been shown to give better relative benefits than an equivalent unbounded algorithm [7].

2.2 Example case library

The examples we use to illustrate the recommendation engineering process are based on an artificial case library in the property domain containing 50 cases. Attributes in the case library are bedrooms (2, 3, 4 or 5), building style (*det*atched, *sem*i-detached or terraced), reception rooms (1 or 2) and location (a, b, c or d). Table 1 shows the ten most similar cases for the incomplete query: beds = 4, style = det, loc = a. The similarity of each case to the target query and the attributes in which it *differs* from the target query are also shown. The 1's in the last three columns show the cases in the standard retrieval set (SRS) for $k = 5$ and in retrieval sets of the same size constructed by BG and R-Net.

Candidates for addition to the R-Net retrieval set are considered in order of decreasing similarity to the target query. The first case to be placed in the retrieval set is the one that is most similar to the target query. Thereafter, a candidate case is added to the retrieval set only if there is no case already in the retrieval set that differs from the target case in the same attributes. Addition of cases to the new retrieval set continues until it reaches the required size or there are no further candidates to be considered.

For the example query, the first case to be placed in the R-Net retrieval set is Case 20. As Case 45 differs from the target query in loc, and Case 39 differs in beds, these are the next two cases to be added to the retrieval set. None of the next 5 cases qualifies for addition to the R-Net retrieval set, as the attributes in which they differ from the target query are either the same as Case 45 (loc) or Case 39 (beds). However, no existing case differs from the target query in the same attributes as Case 47 (beds and loc), so it is the next case to be added to the retrieval set. Similarly, no existing case differs from the target query in the same attributes as Case 46 (style and loc), so it is the last of the 5 cases required to fill the retrieval set.

Table 1. Ten most similar cases for the incomplete query: beds = 4, style = det, loc = a.

No.	beds	style	rec	loc	Sim	difference	SRS	BG	R-Net
20	4	det	1	a	1.00	{}	1	1	1
45	4	det	1	d	0.67	{loc}	1	1	1
39	3	det	1	a	0.67	{beds}	1	1	1
36	4	det	2	b	0.67	{loc}	1	0	0
35	5	det	1	a	0.67	{beds}	1	0	0
21	4	det	1	d	0.67	{loc}	0	0	0
10	4	det	1	c	0.67	{loc}	0	1	0
5	4	det	2	b	0.67	{loc}	0	0	0
47	5	det	2	b	0.33	{beds, loc}	0	1	1
46	4	sem	2	d	0.33	{style, loc}	0	0	1
Similarity:							0.73	0.67	0.60
Diversity:							0.47	0.53	0.63

The similarity and diversity characteristics of the standard retrieval set and the retrieval sets constructed by BG and R-Net are shown in Table 1. The increase in diversity (0.06) provided by BG is exactly balanced by the decrease in average similarity, which equates to a relative benefit of 1. The relative benefit provided by R-Net (1.23) is slightly higher. As we shall see in Section 5, both algorithms are capable of providing much higher relative benefits.

3 THE RETRIEVAL NETWORK

We have seen that the exclusion of cases that differ from the target query in the same attributes as a previously selected case may increase diversity in the R-Net retrieval set. However, building the retrieval set is only part of the recommendation engineering process. In contrast to algorithms that focus on the diversity problem, cases that are excluded from the retrieval set are not discarded by R-Net. Instead, they are organised into groups of cases according to the ways in which they differ from the target query.

Given a target query Q, then for any case C we define:

$$diff(C, Q) = \{a \in A_Q : \pi_a(C) \neq \pi_a(Q)\} \qquad (3)$$

$$dCases(C, Q) = \{C^\circ : diff(C^\circ, Q) = diff(C, Q)\} \qquad (4)$$

We refer to $dCases(C, Q)$ as the *difference* group containing C. It is the set of cases that differ from Q in the same attributes, if any, as C. The difference groups associated with a given query Q can be seen to *partition* the case library; that is, each case belongs to one and only one difference group.

For any cases C_1, C_2, and C we define:

$$diff_Q(C_1, C_2) = \{a \in A_Q : \pi_a(C_1) \neq \pi_a(C_2)\} \qquad (5)$$

$$iCases(C, Q) = \{C^\circ : diff_Q(C^\circ, C) = \phi\} \qquad (6)$$

We refer to *iCases*(C, Q) as the *inseparability* group containing C. It is the set of cases that have the same values as C for all the query attributes. Inseparability of cases with respect to a target query is known to be a common source of imperfect precision in interactive CBR [10]. If two cases are inseparable with respect to a given query, they must also differ from the query in the same attributes.

Figure 1 shows the inseparability and difference groups for a given case C with respect to a target query Q.

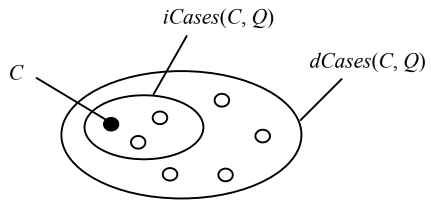

Figure 1. Inseparability and difference groups for a given case C with respect to a target query Q.

We are now in a position to describe how the retrieval network is constructed in R-Net. In Figure 2, Q is the target query, k is the required size of the retrieval set, and *Candidates* is initially the list of all cases, in order of decreasing similarity, that match the query in at least one attribute. Each time R-Net selects a case for addition to the retrieval set, it constructs the difference and inseparability groups for the new case and removes all cases in the difference group from the list of candidate cases.

```
algorithm R-Net(Q, k, Candidates, RetrievalSet)
begin
  RetrievalSet ← ϕ
  while |RetrievalSet| < k and |Candidates| > 0 do
  begin
      C ← first(Candidates)
      RetrievalSet ← RetrievalSet ∪ {C}
      dCases(C, Q) ← {C}
      iCases(C, Q) ← {C}
      for all C° ∈ rest(Candidates) do
      begin
          if diff(C°, Q) = diff(C, Q)
          then dCases(C, Q) ← {C°} ∪ dCases(C, Q)
          if diff_Q(C°, C) = ϕ
          then iCases(C, Q) ← {C°} ∪ iCases(C, Q)
      end
      Candidates ← Candidates - dCases(C, Q)
  end
end
```

Figure 2. Algorithm for recommendation engineering.

A detail not shown in Figure 2 is that for each case C it adds to the retrieval set, R-Net also creates **same**, **like** and **why** links from C to the sets *iCases*(C, Q) - {C}, *dCases*(C, Q) - *iCases*(C, Q) and *diff*(C, Q). As we show in Section 4, the **why** link is used to explain the relevance of a recommended case. The **same** link points to cases, if any, with the same values for all the query attributes as the recommended case. The **like** link points to cases that differ from the target query in the same attributes as the recommended case. Figure 3 shows a fragment of the retrieval network built by R-Net for the example query in the property domain. The **next** link points to the next case in the R-Net retrieval set.

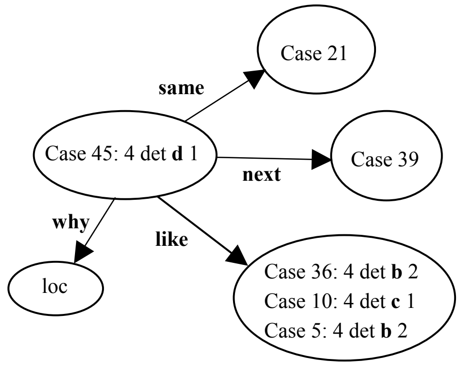

Figure 3. Fragment of the retrieval network for the example query.

The R-Net retrieval set can be seen to include at most one case from each of the difference groups associated with a given query. We now establish an upper bound for the number of difference groups, and hence the size of the R-Net retrieval set.

Theorem 1 *The maximum number of difference groups associated with a given query Q is* $2^{|A_Q|}$.

Proof. For each subset S of the query attributes, there is a corresponding difference group if and only if there is at least one case that differs from Q only in the attributes in S. The result follows from the fact that any set with n elements has 2^n subsets.

R-Net's exclusion of cases that fail to match the target query in at least one attribute reduces the maximum possible size of the retrieval set to $k_{\max} = 2^{|A_Q|} - 1$. For a query involving 5 attributes, $k_{\max} = 31$. In practice, the size of the retrieval set depends on the available cases and may be considerably less than $k_{\max}$.

R-Net has worst-case complexity $O(nk)$, where n is the number of candidate cases and k is the required size of the retrieval set. If necessary in large-scale applications, response times can be improved by reducing either the value of k or the number of candidate cases, for example by requiring a minimum level of similarity to the target query.

4 THE SHOWCASE PROTOTYPE

We now present an implementation of R-Net in a prototype recommender system called *ShowCase*. Figure 4 shows an example dialogue based on the query in Table 1, again with $k = 5$ as the required size of the retrieval set. The example dialogue shows how ShowCase explains the relevance of a recommended case. Such explanations are delivered in response to simple commands. For example, when the user asks why Case 45 is relevant, ShowCase explains that it differs from her query only in location.

The example dialogue also shows how the user can view cases, if any, that have the same values for the query attributes as a recommended case. Of course, such cases may differ from the recommended case in *non-query* attributes. The user can also view cases that differ from her query in the same attributes as a recommended case. Such cases, if any, are immediately available in the retrieval network constructed by ShowCase. When the user asks to see other cases like Case 45, she is shown three cases that also differ from her query only in location, including two that have more reception rooms than Case 45.

ShowCase:	Please enter your query
User:	beds = 4, style = det, loc = a
ShowCase:	The following cases are recommended:
	Case 20: beds = 4, style = det, rec = 1, loc = a
	Case 45: beds = 4, style = det, rec = 1, loc = d
	Case 39: beds = 3, style = det, rec = 1, loc = a
	Case 47: beds = 5, style = det, rec = 2, loc = b
	Case 46: beds = 4, style = semi, rec = 2, loc = d
User:	**why** 20
ShowCase:	This case matches your query exactly.
User:	**why** 45
ShowCase:	This case differs from your query only in loc.
User:	**same** 45
ShowCase:	The following case has the same beds, style, and loc:
	Case 21: beds = 4, style = det, rec = 1, loc = d
User:	**like** 45
ShowCase:	The following cases also differ from your query only in loc:
	Case 36: beds = 4, style = det, rec = 2, loc = b
	Case 10: beds = 4, style = det, rec = 1, loc = c
	Case 5: beds = 4, style = det, rec = 2, loc = b
User:	**why** 39
ShowCase:	This case differs from your query only in beds.
User:	**like** 39
ShowCase:	Another case that differs from your query only in beds is:
	Case 35: beds = 5, style = det, rec = 1, loc = a
User:	**why** 46
ShowCase:	This case differs from your query only in style and loc.

Figure 4. Example recommendation dialogue in ShowCase.

The ShowCase prototype has helped to identify a number of issues to be addressed by further research. For example, its current lack of knowledge of preferences with respect to non-query attributes means that ShowCase is unable to comment, as a human salesperson might, on 'bonus' features of a proposed alternative. The ShowCase dialogue would also benefit from more specific links to alternatives in the retrieval network, on similar lines to the *tweaks* often used in query refinement [2-4]. This would enable the system to anticipate more precisely the directions in which users may wish to explore the solution space (e.g. "Like this, but more bedrooms"), while differing from query refinement techniques in its ability to provide *immediate* access to those pathways. Another difference is that two cases are deemed to be alike in ShowCase if they differ from the target query in the same attributes, whereas in traditional refinement techniques likeness is measured in terms of the underlying similarity measure.

5 EXPERIMENTAL RESULTS

In this section we present an empirical evaluation of R-Net in terms of retrieval-set size and the similarity and diversity of the cases initially presented to the user. The 'Travel' case library on which our evaluation is based is a standard benchmark (*www.ai-cbr.org*) that contains over 1,000 holidays and their descriptions. As incomplete queries are common in practice, our evaluation focuses on 5 of the 8 attributes in the case library, namely: transport (4 values), accommodation (6 values), type (8 values), season (12 values), and region (59 values).

5.1 Retrieval-set size

Our first experiment examines factors that influence the size of the R-Net retrieval set. Our theoretical results show that the maximum possible size is $2^{|A_Q|} - 1$, where $|A_Q|$ is the length of the query. In practice, the size of the retrieval set may depend not only on the length of a query but also on which attributes are included in the query.

For each case in the case library, we generated all possible queries of lengths from 1 to 5 involving the five selected attributes from their values in the case description. For example, the number of queries of length 2 is ${}^5C_2 = 10$. Thus each case gives rise to 32 distinct queries. We then removed the original case from the case library and noted the size of the retrieval set constructed by R-Net for each of the 32 queries. We repeated this process for each of the 1,024 cases in the case library.

For each query length from 1 to 5, Figure 5 shows the maximum, minimum, and average size of the retrieval set constructed by R-Net. The maximum possible size of the retrieval set (Poss) is also shown for each query length.

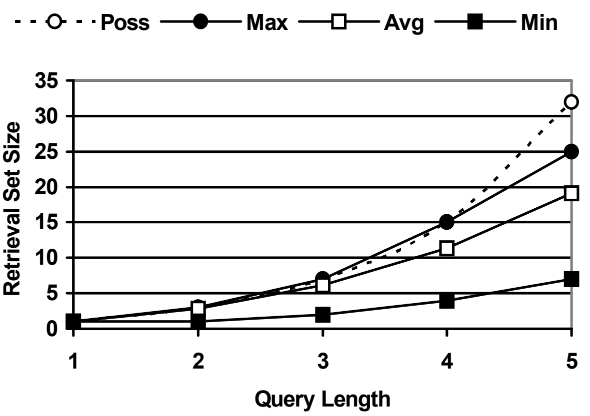

Figure 5. Maximum, average, and minimum size of the R-Net retrieval set for queries of length from 1 to 5.

As might be expected, the retrieval-set size is often much lower in practice than the theoretical maximum. For queries involving all 5 attributes, the minimum size of the retrieval set (7) is only 23% of the theoretical maximum (31). It is interesting to note, though, that the maximum possible retrieval-set size is achieved for all lengths of query except 5. A possible explanation is that queries of maximum length must include region, which has the largest number of values (59). The presence in a query of attributes with large numbers of values can be expected to reduce the number of ways in which cases differ from the query (and hence the size of the retrieval set in R-Net). For example, holidays that differ from a target query in type are also likely to differ in region simply because there are so many regions in the case library.

5.2 Similarity and diversity

Our second experiment examines the trade-off between similarity and diversity in the R-Net retrieval set. Of particular interest is R-Net's performance when the size of the retrieval set is restricted by the available screen size. An important benchmark in our evaluation is BG, an algorithm that focuses on improving the balance between similarity and diversity [7]. Also included in the evaluation is a retrieval strategy (Rand) in which cases in the retrieval set are *randomly* selected from those that match the target query in at least one attribute. Our evaluation focuses on queries involving all five of the selected attributes in the case library, generated as before from the descriptions of actual cases.

For each retrieval-set size k in the range from 2 to 7, we measured the relative benefit, on average, provided by each retrieval strategy; that is, the increase in diversity relative to the standard retrieval set consisting of the k most similar cases divided by the decrease in similarity. The results presented in Figure 6 are based on the average similarity and diversity values over 1,024 queries.

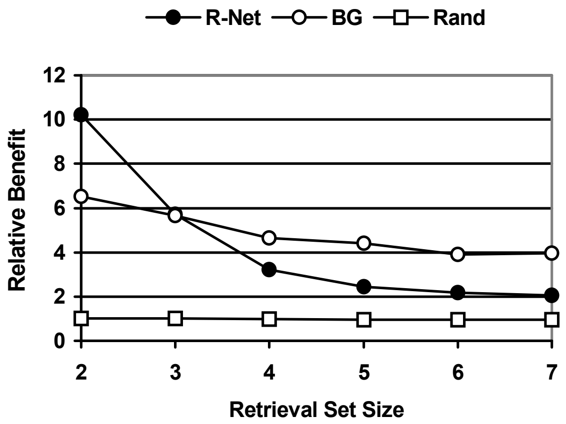

Figure 6. Relative benefits provided by R-Net, BG and Rand for a range of retrieval-set sizes.

Relative benefits in Rand are close to one for all retrieval-set sizes, which means that gains in diversity are exactly balanced by decreases in similarity. BG is outperformed by R-Net (though only slightly for $k = 3$) until the retrieval-set size reaches 4, but gives better results from that point onwards. However, the average increase in diversity provided by R-Net is never less than twice the average decrease in similarity. Interestingly, R-Net equalled BG in terms of average similarity for $k = 2$ but sustained greater losses in similarity from that point onwards. On the other hand, R-Net gave higher average diversity than BG for all retrieval-set sizes.

6 CONCLUSIONS

We have presented an algorithm for recommendation *engineering* called R-Net that moves beyond the standard CBR approach of retrieving the most similar cases. Instead, R-Net builds a retrieval *network*[2] in which the cases it selects for initial presentation to the user are representative of cases that differ from the target query in similar ways. As demonstrated in ShowCase, a major advantage of the approach is the ability to explain why items are recommended in much the same way as a human salesperson might explain their relevance. R-Net also avoids the problem of spurious precision [1] by its use of *qualitative* criteria in the selection of cases (other than the most similar case) for inclusion in the retrieval set.

R-Net differs from algorithms like BG [7] that focus on the diversity problem in that cases excluded from the retrieval set are not discarded. Instead, cases that have identical descriptions to a recommended case, or differ from the user's query in similar ways, are immediately available for inspection by the user. In this way, R-Net provides a solution to the problem that arises when the recommended items (e.g. jobs, rental apartments) are limited in number or sought in competition with other users. In these circumstances, the elimination of good alternatives in the interest of diversity may not be acceptable to users [8]. In terms of the trade-off between similarity and diversity, our results suggest that R-Net is less effective than BG for larger retrieval sets, but may be more effective when the retrieval set is required to be very small.

Our research is currently focusing on the use of knowledge contained in domain-specific similarity measures to improve the quality of dialogue in ShowCase, for example by enabling it to comment on the pros and cons of a proposed alternative.

[2] Retrieval networks are unrelated to Case Retrieval Nets, a memory model in which retrieval is viewed as a process of case completion [11].

REFERENCES

[1] A. Ferguson and D. Bridge, 'Partial orders and indifference relations: being purposefully vague in case-based retrieval', *Proceedings of the 5th European Workshop on Case-Based Reasoning*, 74-85 (2000).

[2] W. Wilke, M. Lenz and S. Wess, 'Intelligent sales support with CBR', in M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard and S. Wess (eds.), *Case-Based Reasoning Technology*, Springer-Verlag, Berlin-Heidelberg, 91-113 (1998).

[3] K.J. Hammond, R. Burke, and K. Schmitt, 'A case-based approach to knowledge navigation', in D.B. Leake (ed.), *Case-Based Reasoning: Experiences, Lessons & Future Directions*, AAAI Press/MIT Press, Menlo Park, CA, 125-136 (1996).

[4] R. Burke, 'A case-based reasoning approach to collaborative filtering', *Proceedings of the 5th European Workshop on Case-Based Reasoning*, 370-379 (2000).

[5] H. Shimazu, 'ExpertClerk: navigating shoppers' buying process with the combination of asking and proposing', *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 1443-1448 (2001).

[6] B. Smyth and P. Cotter, 'A personalised TV listings service for the digital TV age', *Knowledge-Based Systems*, **13**, 53-59 (2000).

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

[8] D. McSherry, 'Increasing recommendation diversity without loss of similarity', *Proceedings of the 6th UK Workshop on Case-Based Reasoning*, 23-31 (2001).

[9] L.K. Branting, 'Acquiring customer preferences from return-set selections', *Proceedings of the 4th International Conference on Case-Based Reasoning*, 59-73 (2001).

[10] D. McSherry, 'The inseparability problem in interactive case-based reasoning', in M. Bramer, F. Coenen and A. Preece (eds.) *Research and Development in Intelligent Systems XVIII*, Springer-Verlag, London, 109-122 (2001).

[11] H.-D. Burkhard, 'Extending some concepts of CBR - foundations of case retrieval nets', in M. Lenz, B. Bartsch-Spörl, H-D Burkhard and S. Wess (eds.), *Case-Based Reasoning Technology*, Springer-Verlag, Berlin-Heidelberg, 17-50 (1998).