Automating the Discovery of Recommendation Knowledge

David McSherry and Christopher Stretch
School of Computing and Information Engineering
University of Ulster, Coleraine BT52 1SA, Northern Ireland
{dmg.mcsherry, ct.stretch}@ulster.ac.uk

Abstract

In case-based reasoning (CBR) systems for product recommendation, the retrieval of acceptable products based on limited information is an important and challenging problem. As we show in this paper, basic retrieval strategies such as nearest neighbor are potentially unreliable when applied to incomplete queries. To address this issue, we present techniques for automating the discovery of recommendation rules that are provably reliable and non-conflicting while requiring minimal information for their application in a rule-based approach to the retrieval of recommended cases.

1 Inroduction

In CBR recommender systems, descriptions of the available products are stored as *cases* in a case library, and retrieved in response to a query representing the user's known requirements. In approaches related to conversational CBR (CCBR) [Aha *et al*., 2001], a query is incrementally elicited in a *dialogue* with the user, often with the aim of minimizing the number of questions the user is asked before an acceptable product is retrieved [e.g., Doyle and Cunningham, 2000; Kohlmaier *et al*., 2001; McSherry, 2003].

Increasing the efficiency of recommendation dialogues is also a major focus of research interest in *critiquing* approaches to navigation of complex product spaces [e.g., Burke, 2002; Reilly *et al*., 2004]. However, an aspect of product recommendation that appears to have received little research attention is the ability of experienced salespersons to make reliable recommendations based on *minimal* information and without engaging in a recommendation dialogue with the customer.

For example, an estate agent may recommend property X to a customer she knows to be interested in a 3-bedroom property in location A without asking about other requirements the customer may have. To be confident about recommending property X, the estate agent must take account of *all* features that may affect its acceptability as well as the relative merits of other available properties. Her recommendation is essentially a prediction that property X is likely to be the *most* acceptable of the available properties regardless of the customer's unknown preferences with respect to attributes other than location and bedrooms.

Providing CBR recommender systems with a comparable ability to make reliable recommendations based on minimal information is the goal that motivates the work presented in this paper. As we show in Sections 3 and 4, basic retrieval strategies such as nearest neighbor are potentially unreliable when applied to incomplete queries because of their failure to take account of all features of a recommended case. To address this issue, we present techniques for automating the discovery of recommendation rules that are provably reliable and non-conflicting while requiring minimal information for their application in a rule-based approach to the retrieval of recommended cases.

In Sections 2 and 3, we present a prototype system for rule-based retrieval of recommended cases and techniques for automating the discovery of *identification* rules that uniquely identify a case from its partial description. In Section 4, we present techniques for the discovery of more reliable recommendation rules that we refer to as *dominance* rules. In Section 5, a theoretical upper bound for the size of the discovered rule sets in our approach is established and confirmed by empirical results based on publicly available datasets. Related work is discussed in Section 6 and our conclusions are presented in Section 7.

2 Recommendation Rules

Given a case library representing a collection of available products, our aim is to automate the discovery of recommendation rules to support rule-based retrieval of recommended products. The rules targeted by our discovery algorithms are of the form **if** Q **then** C, or $Q \rightarrow C$, where Q is a simple query in the form of a list of required features and C is the case that will be retrieved in response to Q. Table 1 shows a small case library in the property domain that we use to illustrate the discussion. Attributes in the example case library are location (loc), style, bedrooms (beds), and reception rooms (RRs).

Table 1. An example case library in the property domain.

Case No.	Loc	Style	Beds	RRs
1	C	semi	5	two
2	C	terraced	4	three
3	B	semi	4	three
4	A	detached	4	two
5	B	terraced	2	three
6	A	detached	3	two
7	C	semi	2	three
8	C	semi	3	two
9	B	detached	3	three

Our discovery algorithms are based on recommendation criteria that give rise to recommendation rules of different types. To ensure that the discovered rules are *non-conflicting*, we insist that for $Q \rightarrow C$ to be a recommendation rule, C must be *strictly* better than any other case according to the underlying recommendation criterion. The recommendation criterion on which we focus in Section 3 is that the recommended case is the *only* case that exactly matches the user's known requirements. A similar or weaker criterion is used in some CBR approaches [e.g., Doyle and Cunningham, 2000; McSherry, 2001]. We will refer to recommendation rules based on exact matching as *identification* rules.

Given a query Q over a subset A_Q of the case attributes A, we refer to $|A_Q|$ as the *length* of the query, and define *exact-matches*$(Q) = \{C : \pi_a(C) = \pi_a(Q)$ for all $a \in A_Q\}$, where for each $a \in A_Q$, $\pi_a(C)$ is the value of a in C and $\pi_a(Q)$ is the value of a in Q. We say that a given query Q is a *sub-query* of another query Q^*, or that Q^* is an *extension* of Q, if $A_Q \subseteq A_{Q^*}$ and $\pi_a(Q) = \pi_a(Q^*)$ for all $a \in A_Q$. We denote the relationship by writing $Q \sqsubseteq Q^*$. For any case C, we refer to the query Q^C such that $\pi_a(Q) = \pi_a(C)$ for all $a \in A$ as the *characteristic* query for C. Clearly for any case C and query Q, $C \in$ *exact-matches*(Q) if and only if $Q \sqsubseteq Q^C$.

Definition 1. *For any case C and query Q, we say that* $Q \rightarrow C$ *is an identification rule if exact-matches*$(Q) = \{C\}$.

The length of a recommendation rule is the length of the query on its left-hand side (LHS). A recommendation rule $Q \rightarrow C$ is *maximally general* (MG) if there is no proper sub-query $Q°$ of Q such that $Q° \rightarrow C$ is also a recommendation rule (of the same type). To ensure that the discovered rules require *minimal* information for their application, we focus on the discovery of MG recommendation rules. As we show in Section 3, the only MG identification rules for Case 6 in our example case library are:

Rule 1. **if** loc = A **and** beds = 3 **then** Case 6
Rule 2. **if** style = det **and** beds = 3 **and** RRs = two **then** Case 6

By focusing on MG rules, we also aim to maximize *coverage* of the product space provided by the discovered rules. However, an important point to note is that an identification rule cannot simply be applied to any query that includes the conditions on its LHS without risking violation of the exact-matching criterion. For example, Rule 1 does not apply for a user looking for a 3-bedroom property in location A if it is also *known* that she prefers a terraced property. However, if $Q \rightarrow C$ is an identification rule and Q^* is any query such that $Q \sqsubseteq Q^* \sqsubseteq Q^C$, then it is clear that $Q^* \rightarrow C$ is also an identification rule.

In *Rubric*, our prototype system for rule-based retrieval of recommended cases, a recommendation rule is applied only to queries that it *covers* in the following sense.

Definition 2. *A recommendation rule* $Q \rightarrow C$ *covers a given query* Q^* *if* $Q \sqsubseteq Q^*$ *and* $Q^* \rightarrow C$ *is also a recommendation rule*.

For example, an identification rule $Q \rightarrow C$ covers any query Q^* such that $Q \sqsubseteq Q^* \sqsubseteq Q^C$. Given a set of recommendation rules, and a query representing a user's known requirements, Rubric checks through the rules and retrieves the case recommended by the first rule that covers the target query. If none of the available rules covers the target query, Rubric simply *abstains* from making a recommendation.

Our rule-based approach to retrieval is related to CBR approaches in which a decision tree is used to guide the retrieval of recommended cases [e.g., McSherry, 2001]. In such a decision tree, each path to a leaf node at which a single case is recommended is an identification rule. However, rule-based retrieval has the potential to provide greater coverage, as a decision tree constructed by standard partitioning methods can have at most one rule for each case, and some of the rules may not be MG.

3 Identification Rule Discovery

Our algorithm for the discovery of MG identification rules, *MGIRules*, is shown in Figure 1. *SubQueries* is a list of all sub-queries, in order of increasing length, of the characteristic query Q^C for a target case C. Each such sub-query is a candidate to appear on the LHS of a discovered identification rule. For any sub-query Q_1 such that *exact-matches*$(Q_1) = \{C\}$, MGIRules adds $Q_1 \rightarrow C$ to the list of discovered rules and eliminates all sub-queries Q_2 of which Q_1 is a sub-query from the remaining list of candidate sub-queries.

To illustrate the approach, Figure 2 shows all sub-queries of the characteristic query (A, det, 3, two) for Case 6 in our example case library. The first identification rule to be discovered is Rule 1 (A, 3). Following the elimination of the underlined sub-queries in Figure 2, the only other identification rule for Case 6 is Rule 2 (det, 3, two).

With each case in turn as the target case, MGIRules can be used to discover all MG identification rules in a given case library. The worst-case complexity of applying MGIRules to all n cases in a case library with k attributes is $O(k \times n^2 \times 2^k)$ if $n \geq 2^k$. If $n < 2^k$, the worst-case complexity is $O(k \times n \times 2^{2k})$.

```
algorithm MGIRules(C, SubQueries)
begin
  Rules ← ϕ
  while |SubQueries| > 0 do
  begin
    Q1 ← first(SubQueries)
    Deletions ← {Q1}
    if exact-matches(Q1) = {C}
    then begin
           Rules ← Rules ∪ {Q1 → C}
           for all Q2 ∈ rest(SubQueries) do
           begin
              if Q1 ⊆ Q2
              then Deletions ← Deletions ∪ {Q2}
           end
         end
    SubQueries ← SubQueries - Deletions
  end
  return Rules
end
```

Figure 1. Algorithm for the discovery of all MG identification rules for a target case.

It is worth noting that an identification rule $Q \rightarrow C$ is excluded by MGIRules only if it has already discovered an MG identification rule $Q° \rightarrow C$ such that $Q° \subseteq Q$. As any query covered by $Q \rightarrow C$ is also covered by $Q° \rightarrow C$, the exclusion of $Q \rightarrow C$ causes no loss of coverage in *Rubric*.

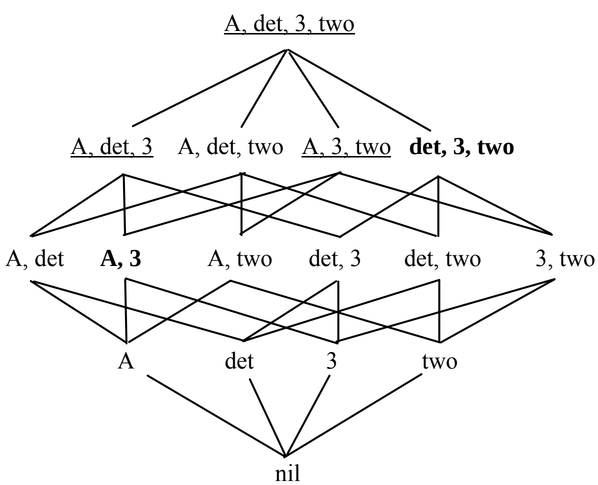

Figure 2. Identification rule discovery for Case 6 in the example case library.

Identification rules have the important advantage that recommendations can be *justified* on the basis that the recommended case exactly matches the known requirements of the user. However, a limitation they share with decision trees is their failure to take account of *all* features of a recommended case that may affect its acceptability. The extent to which this affects their reliability is likely to depend on the importance of the user's *unknown* requirements.

For example, according to Rule 2 in Section 2, Case 6 can be recommended to a user who is known to be looking for a detached property with 3 bedrooms and two reception rooms. But given the importance of location in the property domain, Case 6 is unlikely to be the most acceptable case if the user happens to prefer location B. In fact, Case 9 now looks a better alternative in light of the user's preference for location B. In Section 4, we present techniques for the discovery of recommendation rules that do take account of all features of a recommended case, including those with respect to which the user's preferences are unknown.

4 Dominance Rules

As nearest-neighbor (NN) retrieval is a common approach to product recommendation in CBR, it is natural to consider NN rules as an alternative to identification rules. Given a query Q over a subset A_Q of the case attributes A, the similarity of any case C to Q is typically 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 an importance weight assigned to 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 value of a in Q. When discussing actual similarity scores, we will divide $Sim(C, Q)$ by the sum of all the importance weights to give a *normalized* similarity score. As usual in practice, we assume that for all $a \in A$, $0 \leq sim_a(x, y) \leq 1$ and $sim_a(x, y) = 1$ if and only if $x = y$. We also assume that for all $a \in A$, the distance measure $1 - sim_a$ satisfies the triangle inequality. For any query Q, we define *most-similar*$(Q) = \{C : Sim(C, Q) \geq Sim(C°, Q)$ for all cases $C°\}$.

Definition 3. *For any case C and query Q, we say that Q → C is an NN rule if most-similar*$(Q) = \{C\}$.

As in the case of an identification rule, an NN rule cannot simply be applied to any query that includes the conditions on its LHS. If $Q \rightarrow C$ is an NN rule and Q^* is an extension of Q then there is no guarantee that *most-similar*$(Q^*) = \{C\}$. It is worth noting, though, that $Q^* \rightarrow C$ is an NN rule if $Q^* - Q \subseteq Q^C$. Clearly, any identification rule is also an NN rule.

As we shall see, however, NN rules offer no obvious improvement over identification rules in terms of their reliability when applied to incomplete queries. Again we use the example case library in Table 1 to illustrate the discussion. The importance weights we assign to loc, style, beds, and RRs are 4, 3, 2, and 1. We define the similarity of two values x and y of a numeric attribute a to be $1 - \frac{|x - y|}{\max(a) - \min(a)}$ where, for example, max(a) is the maximum value of a in the case library. Our similarity measure for style is equivalent to applying our similarity measure for numeric attributes to the corresponding number of adjoining properties (det = 0, semi = 1, ter = 2). Finally, our similarity measures for location and reception rooms assign a similarity score of 1 if the two values are the same and 0 if they are not the same.

Returning to our example in Section 3 of a user looking for a detached property with 3 bedrooms and two reception rooms, we can now use Rule 2 as an NN rule to retrieve Case 6 as the recommended case. It can easily be checked that the cases that are most similar to the user's known requirements are Case 6 (0.60) and Case 4 (0.53). But if the user happens to prefer location B, the most similar case in light of this *unknown* preference would be Case 9 (0.90). In fact, Case 6 is likely to be the most acceptable case only if the user happens to prefer location A, as a preference for location C would see Case 8 (0.85) emerging as the most similar case.

As this example illustrates, the reliability of NN rules (and NN retrieval) is open to question when applied to incomplete queries. However, one example of a reliable NN rule is:

Rule 1. **if** loc = A **and** beds = 3 **then** Case 6

It can be seen that the similarity of Case 6 to any query that includes the conditions on the LHS cannot be equaled by any other case regardless of the user's preferences with respect to style or reception rooms. For example, Case 4 reaches its maximum similarity of 0.93 if the user happens to prefer a detached property with two reception rooms, but these additional preferences also increase the similarity of Case 6 from 0.60 to 1.00.

Rules 1 and 2 are positive and negative examples of the type of recommendation rule we refer to as *dominance* rules.

Definition 4. *For any case C and query Q, we say that $Q \rightarrow C$ is a dominance rule if most-similar(Q^*) = $\{C\}$ for all extensions Q^* of Q.*

As well as being more reliable, dominance rules provide more coverage than identification or NN rules. It can easily be seen that if $Q \rightarrow C$ is a dominance rule, then $Q^* \rightarrow C$ is also a dominance rule for any query Q such that $Q \subseteq Q^*$. That is, a dominance rule covers *any* query that includes the conditions on its LHS. An important role in our approach to the discovery of dominance rules is played by the concept of case dominance proposed by McSherry [2003] as a basis for recognizing when recommendation dialogues can be terminated without loss of solution quality.

Definition 5. *A given case C_1 dominates another case C_2 with respect to a query Q if $Sim(C_1, Q^*) > Sim(C_2, Q^*)$ for all extensions Q^* of Q.*

It can be seen that $Q \rightarrow C$ is a dominance rule if and only if C dominates all other cases with respect to Q. McSherry [2003] uses the triangle inequality to show that a given case C_1 dominates another case C_2 with respect to a query Q if and only if:

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

We focus on the discovery of MG dominance rules $Q \rightarrow C$ such that $Q \subseteq Q^C$, where Q^C is the characteristic query for C. As well as reducing the complexity of the discovery process, this ensures that recommendations based on the discovered rules can be justified on the grounds that the recommended case exactly matches *some* of the user's known requirements, and that there is no other case that exactly matches those requirements. It can be seen from the following theorem that our exclusion of dominance rules that do not have this property cannot result in failure to discover dominance rules of the shortest possible length for a given target case.

Theorem 1. *For any dominance rule $Q \rightarrow C$, there exists a dominance rule $Q' \rightarrow C$ of equal length such that $Q' \subseteq Q^C$.*

Proof. Let $Q \rightarrow C_1$ be a dominance rule and let Q' be the query such that $\pi_a(Q') = \pi_a(C_1)$ for all $a \in A_Q$. To establish that $Q' \rightarrow C_1$ is also a dominance rule, it suffices to show that C_1 dominates any other case C_2 with respect to Q'. For any $a \in A_Q$, we know from the triangle inequality that: $1 - sim_a(C_2, Q) \leq 1 - sim_a(C_2, Q') + 1 - sim_a(Q', Q) = 2 - sim_a(C_2, Q') - sim_a(C_1, Q)$. So $sim_a(C_1, Q') - sim_a(C_2, Q') = 1 - sim_a(C_2, Q') \geq sim_a(C_1, Q) - sim_a(C_2, Q)$. As C_1 dominates C_2 with respect to Q and $A_Q = A_{Q'}$, it can now be seen that: $Sim(C_1, Q') - Sim(C_2, Q') \geq Sim(C_1, Q) - Sim(C_2, Q) > \sum_{a \in A - A_{Q'}} w_a(1 - sim_a(C_1, C_2))$. It follows as required that C_1 dominates C_2 with respect to Q'. □

To convert MGIRules to a new algorithm called *MGDRules* for the discovery of all MG dominance rules $Q \rightarrow C$ for a target case C such that $Q \subseteq Q^C$, it is necessary only to replace the condition underlined in Figure 1 by the condition:

C dominates all other cases with respect to Q_1

As in the case of MGIRules, the worst-case complexity of applying MGDRules to all n cases in a product case library with k attributes is $O(k \times n^2 \times 2^k)$ or $O(k \times n \times 2^{2k})$ depending on whether $n \geq 2^k$.

5 Discovered Rules

As might be expected, our algorithms discovered fewer dominance rules (12) than identification rules (24) in the example case library. However, the 12 dominance rules cover 34% of all possible queries in the example product space compared to 28% for the 24 identification rules, and 17% for a set of 9 decision-tree rules of the shortest possible length for each case. In this section, we examine the behavior of our algorithms when applied to case libraries of more realistic size. As the coverage gains provided by dominance rules are measurable only in finite product spaces, our analysis focuses on the number and length of the discovered rules.

It can be seen that for any case C, $Q^C \rightarrow C$ is both an identification rule and a dominance rule provided no other case has the same value as C for every attribute. Two cases with identical descriptions in a product case library are said to be *inseparable* [McSherry, 2002]. At least one rule of each type must therefore be discovered for a target case provided there is no case from which it is inseparable.

We now establish an upper bound for the number of rules that can be discovered by *MGIRules* or *MGDRules* for a given target case.

Theorem 2. *For any case C, the number of MG recommendation rules $Q \rightarrow C$ such that $Q \subseteq Q^C$ can never be more than ${}^{k}C_{[k/2]}$, where k is the number of attributes in the case library and $[k/2]$ is the integer part of $k/2$.*

Proof. First we note that if $Q_1 \rightarrow C$ and $Q_2 \rightarrow C$ are distinct MG recommendation rules such that $Q_1, Q_2 \subseteq Q^C$, then A_{Q_1} and A_{Q_2} are incomparable subsets of A. For example, if $A_{Q_1} \subseteq A_{Q_2}$ then contrary to our assumption $Q_2 \rightarrow C$ cannot be MG. The result immediately follows from Sperner's [1928] proof that the maximum number of incomparable subsets of any set of size m is ${}^{m}C_{[m/2]}$. □

For a case library with 8 attributes, the maximum number of rules that can be discovered by our algorithms for a given target case is ${}^{8}C_{4} = 70$. Table 2 shows the corresponding limits for attribute numbers in the range from 4 to 10. However, we now present empirical evidence which suggests that the discovered rule sets tend to be much smaller in practice than their maximum possible sizes.

Table 2. Maximum rule-set size for a single case.

No. of Attributes:	4	5	6	7	8	9	10
Maximum Size:	6	10	20	35	70	126	252

Both of our experimental case libraries have 8 attributes, and include continuous as well as nominal attributes. Based on the *AutoMPG* dataset from the UCI Repository, our first case library contains the descriptions of 392 automobiles in terms of attributes one might expect to see in a recommender system for previously-owned automobiles (e.g., year, origin, mpg). Our second case library is *Travel* (www.ai-cbr.org), a standard benchmark containing the descriptions of over 1,000 holidays in terms of attributes such as price, destination, and transport.

Figure 3 shows the numbers of rules discovered by our algorithms over all (complete) cases in AutoMPG and Travel, apart from two inseparable cases in Travel for which no rules were discovered. The largest rule-set size for any case (20) is considerably smaller than the maximum rule-set size for 8 attributes (70). On the whole, the results for AutoMPG and Travel are remarkably similar, with fewer dominance rules than identification rules discovered in both case libraries. The average rule-set size of 4 for dominance rules in Travel is based on a total of 4,127 discovered rules.

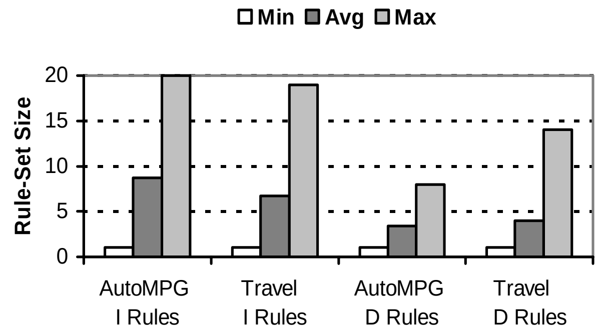

Figure 3. Numbers of discovered identification (I) and dominance (D) rules for each case.

Table 3 shows the *lengths* of the identification and dominance rules discovered by our algorithms in AutoMPG and Travel. As might be expected, the discovered dominance rules are longer on average than the identification rules. In both case libraries, though, the discovered dominance rules provide clear benefits in terms of reducing the number of attributes whose preferred values must be known for a *reliable* recommendation to be made. These benefits are particularly evident in Travel, with reductions in query length of up to 63%, and 38% on average, relative to queries involving all eight attributes.

Table 3. Discovered rule lengths in AutoMPG and Travel.

		Min	Avg	Max
Identification Rules:	AutoMPG	1	2.0	4
	Travel	1	2.9	6
Dominance Rules:	AutoMPG	4	5.6	7
	Travel	3	5.0	7

6 Related Work

Recent work by McSherry [2004a] provides a different perspective on recommendation rule discovery in which the discovered rules (one for each case) are used to describe the *behavior* of an existing recommender system in localized areas of the product space. For example, the discovered rules can be used to identify conditions in which a given product will be recommended by the system, or regions of the product space that are sparsely represented. However, the discovered rules may not be MG and there is no discussion of their possible use for rule-based retrieval.

Burke and Kass [1996] propose a rule-based approach to retrieval in a system for case-based teaching of advertising sales techniques. In *Spiel*, stories relating lessons learned by experienced salespersons are retrieved in response to a student's actions in a simulated sales environment. Retrieval in Spiel is *opportunistic*, *conservative* and *non-mandatory*; that is, stories are retrieved at the system's initiative and only if highly relevant to the student's current situation.

By design, our rule-based approach to retrieval is also *conservative* and *non-mandatory*, though it can easily be combined with NN retrieval of a less strongly recommended case if none of the available recommendation rules covers the user's known requirements. We also propose to investigate its potential role as an *opportunistic* and *complementary* retrieval strategy in recommender systems based on CCBR or critiquing approaches.

Potential benefits include enabling recommender systems to recognize when recommendation dialogues can be safely discontinued without affecting solution quality. Often in critiquing, for example, an initially recommended case is retrieved in response to an initial query entered by the user [e.g., Burke, 2002; Reilly *et al*., 2004]. In this situation, the existence of a dominance rule that covers the user's initial query may be a good indication that a more acceptable case is unlikely to be found no matter how the user chooses to critique the initially recommended case.

Given the importance of *explanation* in recommender systems [e.g., Herlocker *et al*., 2000; McSherry, 2004b], simplifying explanations of why a given product is recommended is another potential benefit in approaches that aim, but cannot guarantee, to minimize the length of recommendation dialogues [e.g., Doyle and Cunningham, 2001; Kohlmaier *et al*., 2001; McSherry, 2003].

7 Conclusions

Aiming to improve the reliability of recommendations based on incomplete queries in CBR recommender systems, we have investigated two possible approaches to the discovery of recommendation knowledge to support a *rule-based* approach to the retrieval of recommended cases. While having the potential to provide greater coverage than decision-tree approaches and enabling recommendations to be easily justified, the *identification* rules discovered by our first discovery algorithm offer no obvious improvement in terms of their reliability when applied to incomplete queries. On the other hand, retrieval based on *dominance* rules is provably more reliable than decision-tree approaches and NN retrieval when applied to incomplete queries in that no competing case can equal the similarity of a recommended case regardless of the user's unknown preferences.

References

[Aha *et al*., 2001] David Aha, Leonard Breslow, and Héctor Muñoz-Avila. Conversational case-based reasoning. *Applied Intelligence*, 14:9-32, 2001.

[Burke, 2002] Robin Burke. Interactive critiquing for catalog navigation in e-commerce. *Artificial Intelligence Review*, 18:245-267, 2002.

[Burke and Kass, 1996] Robin Burke and Alex Kass. Retrieving stories for case-based teaching. In David Leake, editor. *Case-Based Reasoning: Experiences, Lessons & Future Directions*, pages 93-109. AAAI Press, Cambridge, Massachusetts, 1996.

[Doyle and Cunningham, 2000] Michelle Doyle and Pádraig Cunningham. A dynamic approach to reducing dialog in on-line decision guides. In *Proceedings of the Fifth European Workshop on Case-Based Reasoning*, pages 49-60, Trento, Italy, 2000. Springer-Verlag.

[Herlocker *et al*., 2000] Jonathan Herlocker, Joseph Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 241-250, Philadelphia, Pennsylvania, USA, 2000.

[Kohlmaier *et al*., 2001] Andreas Kohlmaier, Sasha Schmitt, and Ralph Bergmann. A similarity-based approach to attribute selection in user-adaptive sales dialogues. In *Proceedings of the Fourth International Conference on Case-Based Reasoning*, pages 306-320, Vancouver, Canada, 2001. Springer-Verlag.

[McSherry, 2001] David McSherry. Minimizing dialog length in interactive case-based reasoning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 993-998, Seattle, Washington, 2001. Morgan Kaufmann.

[McSherry, 2002] David McSherry. The inseparability problem in interactive case-based reasoning. *Knowledge-Based Systems*, 5:293-300, 2002.

[McSherry, 2003] David McSherry. Increasing dialogue efficiency in case-based reasoning without loss of solution quality. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 121-126, Acapulco, Mexico, 2003.

[McSherry, 2004a] David McSherry. Automating the discovery of recommendation rules. In *Proceedings of AI-2004, the Twenty-fourth BCS SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 87-100, Cambridge, UK, 2004. Springer-Verlag.

[McSherry, 2004b] David McSherry. Explanation in recommender systems. In *Proceedings of the ECCBR 2004 Workshops*, pages 125-134, Madrid, Spain. Universidad Complutense Madrid.

[Reilly *et al*., 2004] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Dynamic critiquing. In *Proceedings of the Seventh European Conference on Case-Based Reasoning*, pages 763-777, Madrid, Spain, 2004. Springer-Verlag.

[Sperner, 1928] Emanuel Sperner. Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift*, 27: 544-548, 1928.