

A Case-Based Reasoning View of Automated Collaborative Filtering

Conor Hayes, Pádraig Cunningham

Trinity College Dublin
Conor.Hayes@tcd.ie
Padraig.Cunningham@tcd.ie

Barry Smyth

University College Dublin
Barry.Smyth@ucd.ie

Abstract. From some perspectives Automated Collaborative Filtering (ACF) appears quite similar to Case-Based Reasoning (CBR). It works on data organised around users and assets that might be considered case descriptions. In addition, in some versions of ACF, much of the induction is deferred to run time – in the *lazy* learning spirit of CBR. On the other hand, because of its lack of semantic descriptions it seems to be the antithesis of case-based reasoning – a learning approach *based* on case representations. This paper analyses the characteristics shared by ACF and CBR, it highlights the differences between the two approaches and attempts to answer the question “When is it useful or valid to consider ACF as CBR?”. We argue that a CBR perspective on ACF can only be useful if it offers insights into the ACF process and supports a transfer of techniques. In conclusion we present a case retrieval net model of ACF and show how it allows for enhancements to the basic ACF idea.

1. Introduction

In recent years there has been some discussion on whether Automated Collaborative Filtering (ACF) should be considered a form of Case-Based Reasoning (CBR). It appears that many see CBR-like characteristics in ACF. For instance, entities (users and assets) can be considered to be cases. In addition, some of the reasoning is often deferred to run-time in the lazy manner of CBR. In contrast, others view ACF as a very different approach to CBR, considering ACF to be ‘representationless’ in contrast to the semantic descriptions in a case representation (Smyth & Cotter, 1999) (Balbanovič & Shoham, 1997).

In this paper, we present brief descriptions of case-based reasoning, content-based recommendation and automated collaborative filtering. Then the shared characteristics and differences between ACF and CBR are considered with a view to answering the question “When is it useful or valid to consider ACF as CBR?”

The conclusion of the paper is that this CBR perspective on ACF is useful and this is illustrated in section 5 where we illustrate how ACF may be implemented using a case retrieval net.

2. Characteristics of CBR

CBR is a problem solving technique that reuses previous problem solving episodes in solving new problems. Previous problem solving episodes are stored as cases in a case-base and typically each case has a case specification part and a solution part. In a diagnosis domain for instance, the specification might describe fault symptoms and fault context and the solution describes the cause of the fault. CBR can also be used in situations where this problem-solving vocabulary is not appropriate. More generally, it can be viewed as a means of determining outcomes associated with situation descriptions. With CBR, instead of attempting to model the causal interactions that link outputs to inputs the idea is to retrieve and adapt cases when solving new problems. This is described in Figure 1 where SP is a specification of a problem that needs to be solved, SL is a solution to that problem and FP is some hypothetical First Principles reasoning that would infer the appropriate solution for the problem description SP. The idea in CBR is to avoid having to model this First Principles reasoning by instead retrieving a case with a similar description SP' and adapting the solution to that case (SL') to fit the problem in hand. The implication is that this retrieval and adaptation process is simpler to implement than the First Principles reasoning.

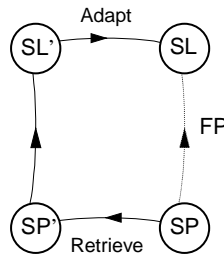


Fig. 1. The case based reasoning process; SP is a specification of a target problem and SL is a solution to that problem, SP' and SL' form a case that is retrieved to solve for this new problem (Cunningham, Finn & Slattery, 1994).

In this view, the special thing about CBR is its ability to avoid the knowledge engineering problem of modeling FP. Instead the problem solving knowledge is stored in the cases.

2.1. CBR as Lazy Learning

In the context of ACF, the other important characteristic of case-based reasoning is that it is a lazy learning technique. This means that induction is delayed to run-time, a factor that can have some important benefits. For instance, in dynamic environments where data is changing continuously, alternative eager approaches have the disadvantage that models (e.g. d-trees or neural nets) can quickly go out of date. Whereas, with the lazy approach of CBR, the induction at run-time reflects the latest

data available. Also the lazy approach of CBR has the advantage that it can model local phenomena well compared to eager techniques that tend to focus on more global models. It will be clear in section 3 that these two advantages of CBR are shared by ACF.

2.2 CBR as Case Completion

The key idea in CBR is that similar problems have similar solutions. This has led to a tendency to view cases as being composed of two parts – a problem specification part and a distinct solution part. This representation is usually adequate for problem scenarios where the problem part is fully specified for a target case and the goal is to retrieve cases with similar problem specifications and use or adapt their solutions for the target problem. This model suggests that the problem specifications are available all at once and that there is no dependency on the order in which problem features are used.

In dialog based CBR there is recognition that a complete problem specification may not be available, and that the order in which the descriptors make themselves available may be important. An example of dialog – based CBR would be the CBR-NET system (Doyle & Cunningham, 2000) where the online user is posed a series of discriminating questions based on the information tendered to date. The system's goal is classification – to recommend a laptop based on the constraints incrementally tendered by the user. In the Nodal system, electronic fault diagnosis is performed by having the tester incrementally perform and submit diagnostic tests. (Cunningham, Smyth & Bonzano, 1998) In both cases the system suggests the next test to perform by carrying out an information theoretic analysis on the subset of cases remaining after submission of the previous test result.

This type of dialogue based CBR is not very different from the conventional model described above. Cases still have distinct specification and solution parts, the difference is that the specification part is filled out during the problem solving process.

Dialog driven CBR

However, for certain types of dialog driven CBR the eliciting of a series of problem features is not only a pre-requisite but is in fact the goal or solution. In these domains the CBR system is a tool to aid problem-solving processes where the objective is to find the next step in a process considering the steps taken to date. An initial case would represent a task or process where very little information is available initially. The CBR system must suggest the steps whereby the task may be completed and the target case filled out.

Burkhard has defined CBR case completion as an elaboration of the target task by collecting case relevant information (Burkhard, 1998). This definition highlights the interactive nature of the case completion process and the requirement for incremental feedback from the real world. The feedback in this scenario is of course an extended example of the *revise* phase of the typical CBR cycle (Aamodt & Plaza, 1994). In case completion the CBR cycle is traversed several times, each period having a

retrieve, reuse and *revise* phase until the target case is complete. The *retain* phase will not be engaged (if at all) until case completion is achieved.

An interesting specialization on case completion is that of *information completion* where the information to complete the case is gathered only from cases in the case base (Lenz et al., 1998) (Waszkiewicz et al., 1999). In both examples the target case represents an instance in time of a process and the goal is to find cases that suggest possible completion scenarios. Whereas in dialog based CBR the user provides the information to complete the cases, in this case completion process the information to complete the cases is inferred from the case-base.

It should be obvious that the typical case representation of problem part and solution part is inappropriate for cases where case completion is the goal. If the objective is to find the next step in a problem solving process with consideration of the steps taken to date it does not make sense to have a specified solution part. There will be no case solution that is distinct from the case specification.

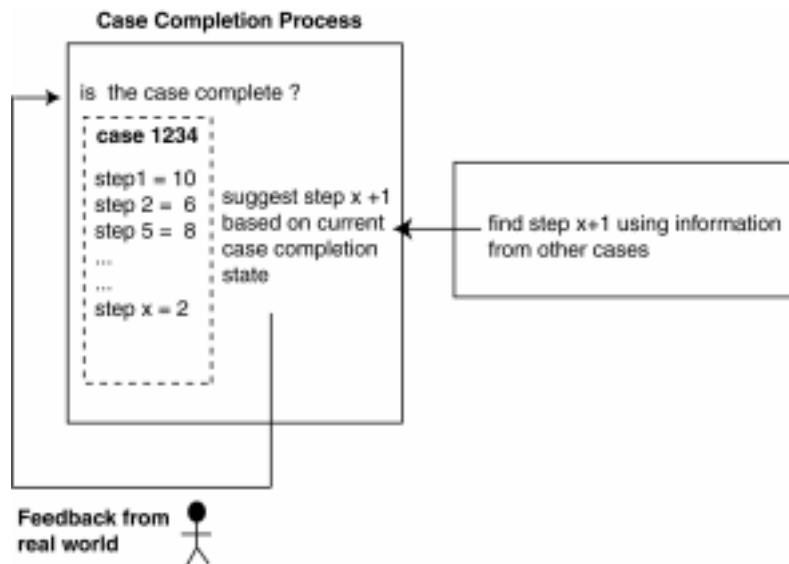


Fig. 2. A general view of the case completion process.

For case completion scenarios Burkhard and Lenz suggest that case representation should not be confined by the standard specification-solution structure. They propose a case as a *view* upon the raw data. They define the term *Information Entity* as the basic unit of information in a case. A case is composed of a number of information entities, one of which may be designated as a solution. This representation caters for cases where information is missing, and where case completion is the goal.

2.3. Content-Based Recommendation

In the next section ACF, a *representation-less* recommendation process, is introduced; before that, we will describe a CBR-like content-based recommendation system that we can use for comparison purposes.

Table 1 shows a case-like description of a film (movie) and Table 2 shows the corresponding description of a user of the recommendation system. In this scenario recommendation is based on how well a film matches a user's profile. In producing recommendations for a user, the matching score for each film in turn would be determined and the highest scoring films not already viewed would be recommended. As will be clear in the next section, this process has advantages over ACF in working well for assets of minority interest or for new assets and users. However, the major drawback is the problem of coming up with appropriate descriptors such as Genre.

Table 1. A case-like description of a film for content based recommendation

4W&1F	
Title:	Four Weddings and a Funeral
Year:	1994
Genre:	Comedy, Romance
Director:	Mike Newell?
Starring:	Hugh Grant, Andie MacDowell
Runtime:	116
Country:	UK
Language:	English
Certification:	USA:R (UK:15)

Table 2. A case-like description of a user's interests

JB-7	
Name:	Joe Bloggs
Preferred Era:	1988 →
Genre:	Thriller, Comedy, War, Romance
Director:	S. Spielberg, F. F. Coppola.
Actors:	Sharon Stone, Sylvester Stallone, Julia Roberts, Keanu Reeves, Liam Neeson, Andie MacDowell
Runtime:	< 150
Country:	UK, US
Language:	English
Certification:	Any

3. Automated Collaborative Filtering

The basic idea of ACF can be shown using Figure 3. In this figure three users have all shown an interest in assets A, B & C (for instance they have all rented videos A B C). This high level of overlap indicates that these users have similar tastes. Further it seems a safe bet to recommend assets D and E to User 1 because they are 'endorsed' by Users 2 and 3 that have similar interests to User 1.

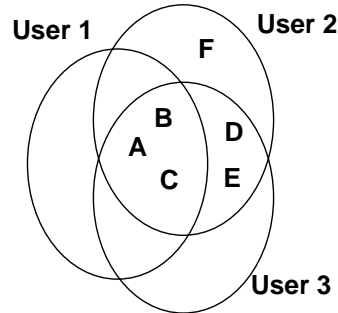


Fig. 3. A Venn diagram showing interests of three users in assets ABCDEF

One of the great strengths of ACF is that, if enough data is available, good quality recommendations can be produced without needing representations of the assets being recommended. The amount of data required depends to some extent on type of data available. In this context, there are two distinct approaches to the ACF idea that are termed *invasive* and *non-invasive*. With the *invasive* approach the user is explicitly asked to rate assets. This is the approach adopted by PTV (ptv.ucd.ie) for instance and clearly the data contains more information (see Table 3). *Non-invasive* data contains less information and can be noisy in the sense that customers may not like some of the items they have used. This can be seen in Table 4, which is a *non-invasive* version of Table 3. The information that User 2 dislikes asset D is lost in the *non-invasive* approach. Because of this data noise and loss of information more data is needed to produce good recommendations with the *non-invasive* approach.

Table 3. Data for use in ACF where users have explicitly rated assets.

	A	B	C	D	E	F	G
User 1	0.6	0.6	0.8			0.8	0.5
User 2		0.8	0.8	0.3	0.7		
User 3	0.6	0.6	0.3	0.5		0.7	0.5
User 4					0.7	0.8	0.7
User 5	0.6	0.6	0.8			0.7	
User 6		0.8	0.8	0.7	0.7		
User 7	0.7	0.5			0.7		
User 8					0.7	0.7	0.8

Table 4. ACF data from Table2 where users have not explicitly rated assets.

	A	B	C	D	E	F	G
User 1	1	1	1			1	1
User 2		1	1	1	1		
User 3	1	1	1	1		1	1
User 4					1	1	1
User 5	1	1	1			1	
User 6		1	1	1	1		
User 7	1	1			1		
User 8					1	1	1

Whether the data available is binary (non-invasive scenario) or contains an explicit rating, the basic structure of the recommendation process will have two distinct phases:

1. First the neighbourhood of users that will produce the recommendations must be determined.
2. Then recommendations must be produced based on the behaviour of these users.

Determining the neighbourhood of users requires a similarity metric and some examples are presented in the next section. This management of neighbourhoods can be done *eagerly* in an offline clustering process. Or it can be done *semi-eagerly* as in GroupLens (Konstan et al., 1997) where a correlations data-base of pair-wise similarities is maintained and used to identify neighbours and build recommendations at run-time. This correlation data-base is updated every 24 hours. In the next subsection we will show the detail of the operation of a lazy ACF system.

3.1. ACF – The Lazy model

(Shardanand & Maes, 1995) evaluate a selection of similarity metrics that may be used for identifying neighbours in ACF. A simple similarity metric could be based on least-squares:

$$\sigma_{UJ} = 1 - \frac{1}{|InCommon|} \sum_{f \in InCommon} (U_f - J_f)^2 \quad (1)$$

where U_f is user U 's rating for asset f . This could be the basis for clustering users in an eager version of ACF or it could be used at run-time to identify neighbours within a threshold. These neighbours can then be used to produce ratings for a series of assets. The expected value of U 's rating for asset x is:

$$U_x = \bar{U} + \frac{\sum_{J \in Raters\ of\ x} (J_x - \bar{J}) \sigma_{UJ}}{\sum_{J \in Raters\ of\ x} |\sigma_{UJ}|} \quad (2)$$

where the *Raters of x* are neighbours that have rated asset x , an asset that would be new to U . In this process an expected value for U 's rating is aggregated from his neighbours. In the aggregation their ratings are normalized using their average rating, \bar{J} and weighted using their similarity to U , σ_{UJ} (Billsus & Pazzani, 1998).

For binary data, the similarity of two users could be measured as:

$$\sigma_{UJ} = \frac{|InCommon(U, J)|}{|U| + |J|} \quad (3)$$

and the rating for an asset could be based on its frequency among neighbours identified with this metric. Clearly binary data is less rich and more data will be needed to produce good recommendations (compare Table 3 & Table 4)

ACF: Criticisms and Simple Extensions

“Amazon.com, the poster child of personalisation, will start recommending needlepoint books to you as soon as you order that ideal gift for your great aunt.”

(www.shorewalker.com)

In fact Amazon.com does not make this mistake because the extreme representation-less view of ACF is unlikely to be pursued in practice. This mistake can be avoided by annotating assets with simple category descriptors in order to allow recommendations to be made in context. Such as simple extension will prevent knitting pattern recommendations *leaking* into a core interest in computing books for instance. This would also address the *latency* problem in ACF whereby new assets cannot be recommended until they have been rated by a number of users.

So the extreme representation-less version of ACF as described above is too restrictive. The addition of basic semantic tags adds considerable value and moves representation of assets and users (customers) more towards CBR-like cases.

ACF & CBR

So a continuum exists with extreme representation-less ACF at one end and CBR at the other. ACF can move toward CBR by attaching semantic descriptors to assets. Also, some *k*-Nearest Neighbour implementations will determine similarity with no reference to the semantics of the case features in a manner that is in the same spirit as the ACF similarity described above.

So the representation issue is not such a defining difference between ACF and CBR. The difference in representation is only one of degree. Each ACF user profile represents a history of that user’s consumption along with either explicit or implicit ratings. Each ACF profile is a record of the users consumption of items to date, the goal being to recommend the next step the user should take in his/her listening or reading behaviour. This temporal perspective on ACF has been neglected to date. As such we suggest that ACF should be viewed as an attempt to model usage patterns where the goal is to suggest the next step in an ongoing process of use.

With these ideas in mind we have come to view ACF as analogous to a dialog driven CBR process. In the next sections we will present our ideas on how closely ACF parallels CBR, and describe our current work on developing a CBR based memory model for ACF. We will describe the benefits of such a system.

4. ACF as dialog driven CBR

ACF is a successful methodology for managing the long-term resource requirements of the online user. The user’s interaction with an ACF based recommender system is usually of a sustained nature and may last from a few minutes to a few years in the case of a successful retail portal like Amazon.com. A target profile is much like a case – it is composed of a number of feature value pairs, where each feature refers to the item consumed and the rating assigned by the user. An ACF case is an incomplete one – it is essentially one row in the user–item matrix and will usually be quite sparsely populated. The key idea in viewing ACF as CBR is to recognize that the goal in ACF is case completion – an incremental elaboration of the user profile based on feedback

given by the user. The system uses the information it has to hand to retrieve similar user profiles and extract completion information for the case profile which is then offered to the user. Negative user feedback may move the user toward a different set of neighbour profiles which are then used to make the next set of recommendations.

So in ACF we have an iterative recommendation process whereby the items recommended for inclusion in the user profile are determined by the feedback to date. The parallel with incremental case completion is obvious. Whilst incremental case completion might generally have a stopping point, in ACF the recommendation cycle is traversed indefinitely, each period having a *retrieve* (similar profiles), *reuse* and *revise* phase. Of course we work with the assumption that there are so many items that the user could not possibly review each, hence the recommendation cycle is repeated continuously.

The *adaptation* phase in the ACF system is the actual process of information completion, of choosing those components of neighbour profiles that are suitable for recommendation to the target user. As described in section 3.1 ACF uses weighted majority voting or the weighted aggregation of the scores of nearest neighbours to produce the recommended components for the target user profile. This may easily be compared to simple CBR adaptation techniques such as producing a solution by weighted majority voting of the k -nearest neighbours.

5. Using Case Retrieval Nets for ACF

In the previous section we examined how the typical ACF approach is similar to a lazy case completion process. Since the ACF process involves an extended dialog with many users the amount of information in the system will increase very quickly. Hence, it is appropriate to examine whether we should use our data to build eager structures that may work in conjunction with the lazy ACF process. In this section we introduce our design of an ACF memory model based on Case Retrieval Nets (CRNs). We will firstly describe basic case retrieval nets, then our implementation of them in an ACF context and finally we will describe why CRNs are an appropriate memory structure for extending ACF systems.

5.1 Case Retrieval Nets

Our initial investigation of CRNs was motivated by their use in domains where missing case information is typical and where the ‘problem–solution’ case representation is inappropriate. Indeed Lenz suggests that CRNs are designed specifically for information completion processes (Lenz et al., 1998). Secondly, new cases and case features can be added without having to rebuild the memory structure which is necessary for ACF where the user ‘case base’ and the case profiles themselves are growing. Firstly, we give a brief summary of CRNs. The reader is referred to Lenz (Lenz, 1999) for a formal definition of the CRN model.

A CRN is a memory model that builds a net instead of a tree from the case base. It uses organizational features derived from associative memory structures (Shank, 1982) and a spreading activation process similar to that used in connectionist

models. In contrast to the latter, however, all the nodes and arcs in the net should have precise meaning.

The components of a basic case retrieval net are as follows:

Information Entities nodes: CRNs have a node for each Information Entity (IE) observed in the domain (see fig. 4). Information Entities are any basic knowledge item such as a particular attribute-value pair. Cases will typically be made up of a number of IEs. In order to facilitate the spreading activation process IE nodes are connected by *similarity arcs*.

Case nodes: each case in the case base has a case node which is reachable from its constituent IE nodes via *relevance arcs*.

By varying the arc weights we can express differing degrees of similarity and relevance between nodes. Case retrieval using a CRN has three stages:

1. *Initial Activation:* this involves activating the IEs in the query. Given the query, the initial activation is determined for all IE nodes.
2. *Similarity Propagation:* the second step involves incrementally propagating the activation through the net of IEs. The amount of activation depends on arc weights.
3. *Relevance Propagation:* the final step entails collecting the achieved activation in the associated case nodes. This is done using the relevance arcs which connects each case node to its constituent IE nodes.

The result of the retrieval process is a set of cases ranked in order of decreasing activation.

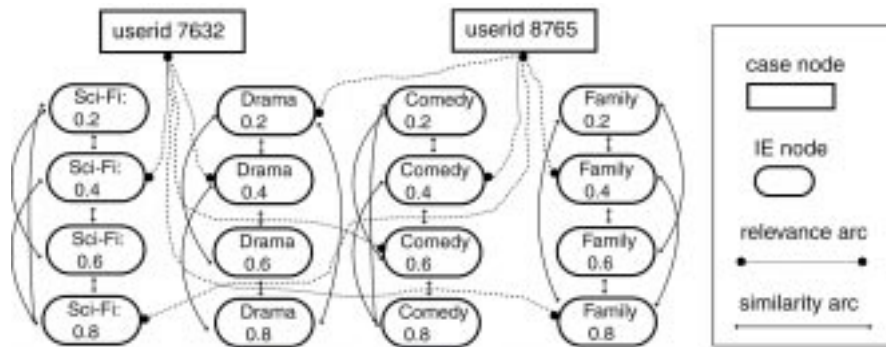


Fig. 4. A portion of a case retrieval net showing two case nodes and a selection of IE nodes. This illustrates the type of CBR data collected during the ptv project (Smyth & Cotter, 1999);

5.1 ACF and CRNs

If we view ACF profiles as incomplete cases as described in section 4 it is clear how we may implement a CRN based ACF system in which the Information Entities nodes represent item-value pairs and the case nodes represent the user profile identifiers. Figure 5 illustrates a simple example using a portion of a user profile in the *smart radio* domain (Hayes & Cunningham, 2000). What makes the spreading

activation a little more complicated in ACF is that whereas in the CBR model similarity arcs tend to extend between IE nodes in which the feature descriptor part is the same, there is no reason why this should be the case in the ACF model. A similarity arc may extend between two different item types such as items 1959 and 1968 in figure 5.

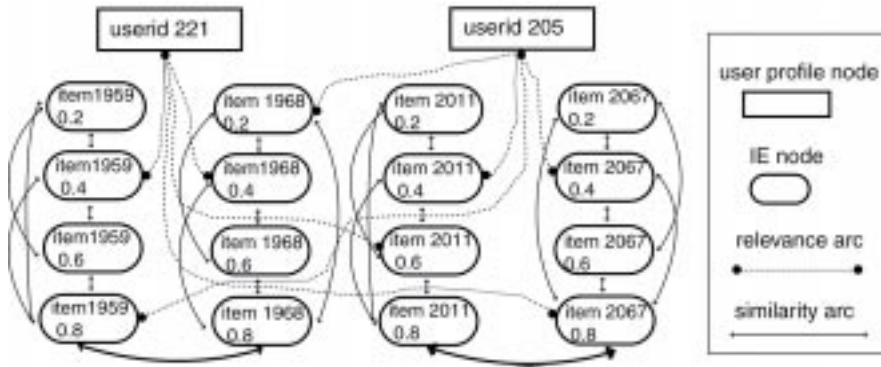


Fig. 5. ACF implemented as a CRN

However, figures 4 and 5 also illustrate the weakness of the basic CRN model we have outlined, namely the requirement for explicit similarity arcs between IE nodes. Where the number of nodes is large it may be infeasible to define similarity arcs in such a way. One way to address this is to replace direct connections to IE nodes with connections to a set of more abstract nodes representing concepts or micro-features in the domain. There may be some descriptive information already on hand. For instance, in the *smart radio* data there are some basic attributes available such as *artist*, *album* and *genre*. However, most genre category information is simply not discriminating enough (i.e. rock, pop), though some such categories such as *folk* or *electronic* may be a little better. This *free* information may form a layer of micro-feature nodes below the IE layer, linking several IE nodes in one go. This schema has a direct corollary in Lenz's description of micro-features in the Cabatta system (Lenz, 1999).

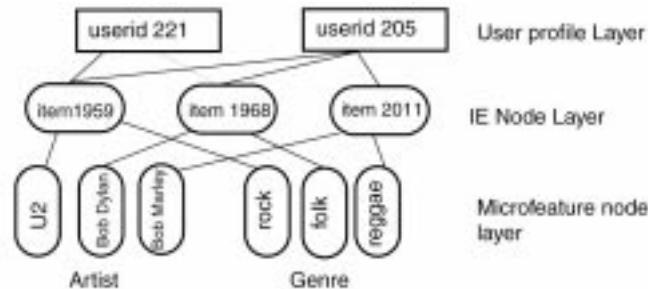


Fig. 6. Example of micro-features describing relationships between different IE nodes

5.2 Embedding discovered knowledge in CRNs

The standard ACF system learns in a very simple way – it collects use-data and lazily produces new recommendations. Since the relationship is an extended one, a smarter system would seek to learn the patterns informing a user’s resource requirements at any time and to determine the relationships between assets that are not *a priori* given.

As we suggested at the top of this section it is possible to find rules and relationships between information entities in an ACF system using knowledge discovery techniques. The relatively *expensive* information found using these techniques may be incrementally added to the CRN memory structure without having to rebuild it.

We can do this using an extension to the basic case retrieval net model called *conceptual case retrieval nets* (Lenz, 1999). The key idea is to introduce additional nodes into the net which do not represent actual attribute-value pairs, but more abstract domain knowledge. These concept nodes allow us to link several IE nodes as shown in fig 8. During the retrieval process the initially activated nodes pass activation according to their *similarity functions* as usual, but also pass activation to connected concept nodes, which in turn distribute activation to connected IE nodes. As such we begin to address the problem in CRNs of defining similarity relations between each pair of IE nodes. By reducing the explicit number of similarity relations between IE nodes we reduce maintenance and computational expense.

Some domain knowledge may in certain circumstances be freely available as mentioned in the previous section, but more generally it will require expert analysis of the domain to find deeper structures. However, it may be infeasible to have an expert keep track of the shifting concepts informing a particular domain. Where we have a lot of use-data it makes sense to employ well established knowledge discovery techniques to mine the concepts.

Fig. 7 presents three simple examples of concepts mined from *smart radio* data. These examples which relate artists in the data set were produced using Cobweb, Fisher’s incremental concept formation algorithm (Fisher, 1987).

<p>C410 ['Martyn, John', 'Orton, Beth', 'Guthrie, Arlo'] C438 ['The Beatles', 'Raitt, Bonnie', 'Fitzgerald, Ella', 'Simone, Nina', 'Lennox, Annie', 'Amos, Tori'] C1066 ['Hill, Lauryn', 'Scott-Heron, Gil', 'Van Morrison', 'Getz-Gilberto', 'Evans, Bill']</p>

Fig. 7. An example of some basic artist concepts found using Fisher’s Cobweb algorithm on smart radio data.

A second source of information which can be used in the CRN model are *association rules*. Typically used in market basket analysis, the key idea is to find interesting relationships between different items in a given data set. By mining association rules we can build similarity arcs between associated IE nodes, thus boosting the activation of connected cases. This activation would not have spread to these cases without the use of discovered association rules.

In the *smart radio* domain (Hayes & Cunningham 2000), for instance, users build music playlists and are in turn recommended other people’s playlists. Running the

Apriori algorithm on these lists allows us to find association rules between tracks and between artists (Agrawal et al. 1996).

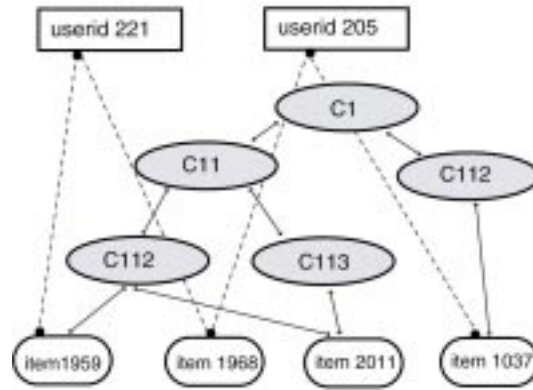


Fig. 8. Embedding concepts in an ACF CRN. The figure illustrates a hierarchal concept structure as produced by a concept formation algorithm like Cobweb.

5.3 Benefits for ACF

A case retrieval net memory model caters to the unbounded nature of the ACF process. Initially, we can make recommendations using a sparse memory structure similar to that shown in figure 5. As the amount of information in the system grows we can incrementally derive rules and concepts which we can add to the case retrieval net as shown in figure 8. This eagerly derived information is used to augment the essentially lazy ACF recommendation process, particularly in areas where it is a weak such as the latency period mentioned in section 3. This sort of additional information is not fixed but shifting and would need to be run periodically to keep up with the level of activity in the community. Secondly, we surmise that this additional information will allow us to reduce the number of user profiles in the case retrieval net. Choosing which profiles are members would involve developing a theory of ACF *coverage*. In the concluding section we address some of the issues involved.

Finally, Lenz et al. propose that the CRN caters for a bottom up search which is appropriate in a situation where information is missing. Apart from the difficulties encountered with missing information with an inductively derived model, they argue that cognitive science would tend to support a memory model that is *reconstructive* rather than purely search oriented: i.e. humans tend to start with a sparse representation and gather more information until a clearer picture is built. (Lenz et al., 1998) These ideas can be found in memory models proposed for early CBR (Schank, 1982) (Kolodner, 1993). Whereas ACF has often been regarded as being on the periphery of AI, using a CRN approach pushes it much closer to the center.

6. Conclusions and further work

We have put forward the idea that ACF is very similar to dialog-driven CBR. Both approaches are involved in the process of case completion. We have then sought to unify both approaches by suggesting Case Retrieval Nets as a suitable memory structure for ACF. Since ACF involves an extended dialog with the user, CRNs offer the advantage of allowing us to incrementally add eagerly learned rules and content.

However, the amount of raw data generally required by ACF may complicate matters. The general claim for a reduction in knowledge engineering in CBR assumes that the case instances are readily available. A CBR system may require careful authoring of its cases which can require a substantial amount of 'expensive' knowledge (Aha, 2001, Aha 1997). If we are to consider ACF in CBR terms, we must examine whether we can author or 'prune' user profiles, and whether it is possible to apply techniques of case coverage to remove 'redundant' profiles.

Case base maintenance is an important issue in CBR both for reasons of efficient retrieval and to ensure the problem space is sufficiently covered. A key research issue is whether we can choose component user profiles of the CRN using case coverage criteria (Smyth & McKenna 1998). Thus, though all case profiles are kept in a case-base, the subset that best covers the user space *and* the asset space would be used in the Case Retrieval Net. We would need also to take into account a measure of which profiles consistently produce good recommendations.

The idea of coverage in ACF is complicated by the fact that user profiles themselves are growing. We are left with the option to leave the 'raw data' of the profile alone and lazily make recommendations, or to seek to reduce this data into cases profiles. This would necessitate reducing the ACF profile by summarising some of the data, by finding key items that will differentiate one user group from another.

Current coverage models assume a single-shot CBR system. A case either covers a problem or it does not. There is no partial coverage. Now in many CBR scenarios we have multiple-cases retrieved to solve a problem much as we do in ACF. A CBR coverage model that takes into account partial coverage situations would of course fit the ACF model very well.

References

- Agrawal, R., Manilla, H., Srikant, R., Toivonen, H., Verkamo, A.I. (1996) Fast discovery of association rules in *Advances in Knowledge Discovery and Data mining*, pp. 307-328, eds. Fayyad, U.M., Piatetsky-Shapiro, G, Smyth, P., Uthurusamy, R. AAAI/MIT Press 1996.
- Aamodt, A. and Plaza, E., (1994). Case Based Reasoning: foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39-59
- Aha, D. W. (1997). A proposal for refining case libraries. In R. Bergmann & W. Wilke (Eds.) *Proceedings of the Fifth German Workshop on CBR (TR LSA-97-01E)*.
- Aha, D., (1998) Reasoning and Learning: The Lazy-Eager Dimension, Invited Keynote Talk at EWCBR 1998, <http://www.aic.nrl.navy.mil/~aha/>
- Aha, D., (2001). Conversational Case based Reasoning in *Applied Intelligence* (14:1), special issue on "Interactive CBR", Kluwer
- Balbanovic, M., Shoham, Y., (1997) Fab: Content-Based Collaborative Recommendation, *Communications of the ACM*, Vol. 40, No. 3, pp66-72.

- Billsus, D., & Pazzani, M.J., (1998) Learning Collaborative Information Filters, in Proceedings of AAAI Workshop on Recommender Systems. AAAI Press, 24-28.
- Burkhard, H-D., (1998) Extending Some concepts of CBR – Foundations of Case Retrieval Nets, in Case Based Reasoning Technology from foundations to applications, eds Lenz, M., Bartsch-Spörl B., Burkhard, H-D., Wess, S. , LNAI 1400, pp17 –50, Springer-Verlag.
- Cunningham, P., (1998) CBR: Strengths and Weaknesses, in Proceedings of 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, eds A. P. del Pobil, J. Mira & M. Ali, Lecture Notes in Artificial Intelligence 1416, Vol. 2, pp517-523, Springer Verlag.
- Cunningham P., Bonzano, A., (1999) Knowledge Engineering Issues in Developing a Case-Based Reasoning Application, Knowledge Based Systems Vol. 12, pp372-379.
- Cunningham P., Finn D., Slattery S., (1994) Knowledge Engineering Requirements in Derivational Analogy in Topics in Case-Based Reasoning, Lecture Notes in Artificial Intelligence, S. Wess, K-D Althoff, M. M. Richter eds., pp234-245, Springer Verlag.
- Cunningham, P., Smyth, B., Bonzano, A., (1998) An incremental retrieval mechanism for case-based electronic fault diagnosis, Knowledge-Based Systems (11)3-4, pp. 239-248
- Doyle, M., Cunningham, P., A Dynamic Approach to Reducing Dialog in On-Line Decision Guides, 5th European Workshop on Case-Based Reasoning (EWCBR 2000), Lecture Notes in Artificial Intelligence, E.Blanzieri, L. Portinale (eds.), pp49-60, Springer Verlag.
- Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2, 139-172.
- Hayes, C., Cunningham, P., (2000) Smart Radio - Building community based music radio, in Applications and Innovations in Intelligent Systems VIII, eds., Macintosh, A., Moulton, M., Coenen, F. , BCS Conference Series, Springer-Verlag.
- Kolodner , J.L., (1993) Case Based Reasoning. Morgan Kaufmann, San Mateo.
- Konstan, J.A., Miller, B.N., Maltz, M., Herlocker, J.L., Gordon, L.R., & Riedl, J., GroupLens: Applying collaborative filtering to Usenet News. Communications of the ACM, Vol. 40, No. 3, pp77-87.
- Lenz, M., Auriol E., Manago M., (1998) Diagnosis and Decision Support, in Case Based Reasoning Technology from foundations to applications, eds Lenz, M., Bartsch-Spörl B., Burkhard, H-D., Wess, S. , Lecture Notes in Artificial Intelligence 1400, pp17 –50, Springer-Verlag.
- Lenz, M., (1999) Case Retrieval Nets as a model for building flexible information systems. PhD dissertation, Humboldt Uni. Berlin. Faculty of Mathematics and Natural Sciences.
- Richter, M. M. (1998). Introduction (to Case-Based Reasoning). in Case-based reasoning technology: from foundations to applications, Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D. & Wess, S. (eds.) (1998). Springer-Verlag, LNAI 1400, pp1-16.
- Schank, R.C., (1982) Dynamic Memory: A Theory of Learning in Computers and People. Cambridge University Press, New York.
- Shardanand, U., and Mayes, P., (1995) Social Information Filtering: Algorithms for Automating ‘Word of Mouth’, in Proceedings of CHI95, 210-217.
- Smyth, B. & McKenna E., (1998) Modeling the competence of case-bases. In *Advances in Case-Based Reasoning*: Proceedings of the Fourth European Workshop on Case-Based Reasoning, pages 196-207. eds.: Barry Smyth and Pádraig Cunningham. Springer-Verlag, Berlin, Germany, September 1998
- Smyth, B. & Cotter, P., (1999) Surfing the Digital Wave: Generating Personalised TV Listings using Collaborative, Case-Based Recommendation, in Proceedings of 3rd International Conference on Case-Based Reasoning eds K-D. Althoff, R. Bergmann, L. K. Branting, Lecture Notes in Artificial Intelligence 1650, V pp561-571, Springer Verlag.
- Waszkiewicz, P., Cunningham, P., Byrne, C., (1999) Case-based User Profiling in a Personal Travel Assistant, User Modeling: Proceedings of the 7th International Conference, UM99, Judy Kay, (ed).pp. 323-325, Springer-Wien-New York.