# Using Collaborative Filtering Data in Case-Based Recommendation

**Derry O' Sullivan** and **David Wilson** and **Barry Smyth**
Smart Media Institute
University College Dublin
{dermot.osullivan,david.wilson,barry.smyth}@ucd.ie

### Abstract

In the context of PTV, an applied recommender system operating in the TV listings domain, we are examining the potential benefits in merging case-based and collaborative filtering (CF) recommendation techniques by developing case-based reasoning (CBR) methods that employ collaborative filtering style ratings profiles directly as cases. Doing so presents a number of challenges, both in applying a case-based perspective to collaborative filtering, and in addressing the sparsity problem that plagues many collaborative filtering systems. This paper expands on earlier CBR views of collaborative filtering, identifies problems and opportunities for similarity maintenance therein, and proposes and evaluates methods for mining and applying new similarity knowledge.

## Introduction

In the context of recommender systems, case-based and collaborative filtering techniques have been, at the same time, viewed as both complimentary and contrasting. They can both be seen as lazy similarity-based reasoning techniques. Case-based methods generate recommendations by prioritizing cases that are similar to ones that the target user has preferred in the past, while collaborative filtering methods prioritize cases that have been liked by users similar to the target user. However these techniques differ significantly in the way that similarity is assessed and the type of data on which the similarity computation is based. Case-based methods rely on rich feature-based representations and sophisticated similarity metrics that make use of heterogeneous similarity measures in order to deal with the various features that can make up a case. Collaborative filtering methods, in contrast, make use of very simple correlation-based similarity techniques to measure the similarity between user profiles that are typically little more than ratings lists; that is, content item identifiers plus an associated preference rating.

In this paper we are interested in developing a case-based recommendation technique that uses ratings-based profiles (a la collaborative filtering) directly as cases. However, this presents a number of challenges that are side-effects of the so-called *sparsity problem* that plagues many collaborative filtering systems. The sparsity problem tells us that, on average, two users are unlikely to have rated many of the same items and so there will be little direct overlap between their profiles. This is problematic when it comes to case retrieval, because it means that there is no direct way to measure the similarity between two profile cases unless we have access to similarity knowledge that allows us to compare non-identical profile items. Unfortunately this similarity knowledge is usually hard to acquire.

Moreover, even when we do have access to similarity knowledge it is not clear how it should be used to determine case similarity. Ordinarily, in CBR systems, the case retrieval step is facilitated by the so-called *alignment assumption* which states that two cases can be compared on a feature by feature basis. But when comparing profile cases it may be possible to relate individual features (rated items) in the candidate case to many features in the target case; in other words there may be no clear one-to-one correspondence between items.

In this paper we address both of these challenges by describing how similarity knowledge can be mined from the profile cases themselves and by advancing a novel similarity metric for comparing non-aligned cases. We demonstrate the effectiveness of these ideas through an experimental evaluation and argue that our technique has broader implications for the CBR community as a whole.

## Setting the Scene

PTV is a deployed recommender system operating in the TV listings domain that combines case-based and collaborative filtering techniques by interleaving the recommendations (TV programmes) from each approach to produce personalized TV guides for each individual user (Smyth & Cotter 2001). The key to PTV's personalization facility is an accurate database of interactively acquired user preference profiles that contain collaborative filtering style ratings lists. These are employed directly in the collaborative filtering component and by transformation to a content summary *profile schema* for matching in the case-based component. We would like to improve recommendations in the overall system by taking this combination a step further with a case-based view of the collaborative component itself. Within this view, we have identified opportunities for maintaining and improving collaborative recommendations and for further developing the relationship between CBR and CF. We are mindful, however, that the collaborative filtering component is desirable, in part, because it can provide diverse and high-quality recommendations with minimal knowledge-engineering effort. One of the goals of this work is to strike a good balance between improving the collaborative recommendations and the amount of development overhead in doing so.

## A CBR Perspective on CF

Recent CBR research has started to investigate the relationship between CBR and CF (e.g., (Hayes, Cunningham, & Smyth 2001; Burke 2000)). Hayes et al. (2001) forward a view of Automated Collaborative Filtering as a lazy CBR case-completion process. We adopt this view here, emphasizing two important differences from typical CBR practice.

First, there is no case solution distinct from case specification. This follows research developments in CBR from dialog-based/conversational systems (e.g., (Doyle & Cunningham 2000; Aha, Breslow, & Muñoz-Avila 2001)) to case/information-completion systems (e.g., (Burkhard 1998; Lenz, Auriol, & Manago 1998)). In dialog-based systems, system-guided user interaction is used to fill out the problem specification during problem-solving, but the goal remains to find a distinct solution based on the problem specification. Case-completion systems, on the other hand, are dialog-based systems in which the elicitation of problem features is both the means and the end. For example, incrementally filling in the next aspect of a design may be supported by reasoning from the current partial context of the design so far (Leake & Wilson 2001). Cases, then, are particular points in an incremental development process, and the case-based reasoning cycle is applied repeatedly to support their refinement, which is very much in the spirit of case life-cycle models (Minor & Hanft 2000). We also note that some case life-cycles may be expected to end, as in completing a design, while other life-cycles may be expected to persist indefinitely, as in recommendation.

Second, ratings-based profiles cases do not conform to the typical structure of parallel features with corresponding heterogeneous similarity measures. This is a direct result of the sparsity typical in collaborative similarity spaces, and we discuss the ramifications in the following section. Thus the *alignment assumption* that two cases can be compared on a feature by feature basis does not necessarily hold in such systems.

As Hayes et al. point out, the key idea in a CBR view of CF lies in recognizing that the goal in CF is case completion, incrementally elaborating user profiles based on system reasoning support and user feedback.

## The Similarity Coverage Problem

Taking a CBR view of CF, we must begin to address weaknesses in the underlying technique, in particular the *sparsity problem*. In collaborative filtering, the sparsity problem tells us that on average two users are unlikely to have rated many of the same items and so there will be little direct overlap between their profiles. From a CBR perspective, there may be no direct way to measure the similarity between two such profile cases. The tacit assumption in CBR has generally been that the similarity metric will cover all potential retrieval situations. It may do so poorly, necessitating for example, learning appropriate feature weights, but it would provide complete coverage. This is not the situation with profile cases, some of which may not be comparable at all, and it presents us with a *similarity coverage problem*, akin to the case-base coverage problem (Smyth & McKenna 1998) that has received a great deal of recent attention.

Addressing the similarity coverage problem is a maintenance issue for CBR systems, directed at the similarity knowledge container (Wilson & Leake 2001). In order to maximize similarity coverage, we need to employ techniques to derive and extend similarity knowledge. In our case, we need to gain access to similarity knowledge that allows us to compare non-identical profile items.

## Mining Similarity Knowledge

There are many automated techniques that could be used to derive various sorts of similarity knowledge. The initial approach we have chosen is to apply data mining techniques (see (Hipp & Nakhaeizadeh 2000) for an overview)), in particular the *Apriori* algorithm (Agrawal *et al.* 1995), to extract association rules between programmes in PTV user-profile cases. By discovering relationships between programmes beyond simple direct overlap, we may be able both to cover more potential profile matches and to make more informed recommendations. For example, a person that likes *X-Files* and *Frasier* would not normally be comparable to a person that likes *Friends* and *ER*, but mining a relationship between *Frasier* and *Friends* would provide a basis for profile matching.

The association rules are of the form $A \Rightarrow B$, where $A$ and $B$ are sets of items (television programmes). In data mining terms, whenever a transaction (case) $T$ contains a certain itemset (set of programmes) $A$, then the transaction probably contains another itemset $B$. The probability that a given rule holds, *rule confidence*, is the percentage of transactions containing $B$ given that $A$ occurs:

$$P(B \subseteq T | A \subseteq T) \tag{1}$$

The *support* of an itemset $A$ is defined as the fraction of transactions supporting $A$ with respect to the entire database. The support of a rule $A \Rightarrow B$, then, is the probability that both itemsets occur together in a

| Rule | Support | Confidence |
|---|---|---|
| Friends $\Rightarrow$ Frasier | 12% | 25% |
| Friends $\Rightarrow$ ER | 14% | 37% |
| Frasier $\Rightarrow$ ER | 10% | 22% |

Table 1: Selected Rules for PTV

| | Friends | Frasier | ER |
|---|---|---|---|
| Friends | 1 | .25 | .37 |
| Frasier | - | 1 | .22 |
| ER | - | - | 1 |

Table 2: Example Item Similarity Matrix

transaction:

$$support(A \Rightarrow B) \quad = \quad P((A \cup B) \subseteq T) \tag{2}$$

The measure of rule confidence is related to support, and can be computed as follows:

$$confidence(A \Rightarrow B) \quad = \quad \frac{support(A \cup B)}{support(A)} \tag{3}$$

In mining association rules, the confidence and support values are used to constrain exponentially large candidate rule sets by setting thresholds.

## Item Similarities

Treating PTV user profiles as transactions and the rated programmes therein as itemsets, the Apriori algorithm can be used to derive a set of programme-programme rules and associated confidence levels. We have limited this initial phase of the work to rules with single-programme antecedents and consequents, leaving multiple-programme itemsets as future work. Table 1 shows some sample rules that were generated by running Apriori on our PTV data set.

The confidence values are taken as probabilities and used to fill in a programme-programme similarity matrix, as shown in Table 2, which provides the additional similarity knowledge necessary to compare non-identical profile items.

Since the matter of additional similarity coverage rests in populating the matrix as densely as possible, two natural extensions suggest themselves. First, the directly generated rules can be chained together ($A \Rightarrow B$ and $B \Rightarrow C$ imply $A \Rightarrow C$) to provide indirect programme relationships. A choice has to be made in how the indirect rule confidence will be calculated (e.g., minimum, maximum, or some combination of the confidences in the potential paths); in this paper we assume a multiplicative model. Second, while it is not logically implied, we would like to see whether rule symmetry (e.g., *Friends$\Rightarrow$Frasier* supporting *Frasier$\Rightarrow$Friends*) could be exploited to extend coverage. Rule similarity knowledge that is generated by Apriori, we refer to as *direct*, and additional derived knowledge as *indirect*.

# A New Recommendation Strategy

The availability of item similarity knowledge facilitates a new type of similarity-based recommendation strategy that combines elements from case-based and collaborative filtering techniques. It facilitates the use of more sophisticated CBR-like similarity metrics on ratings-based profile data, which in turn make it possible to leverage indirect similarities between profile cases, and so generate improved recommendation lists. This new recommendation strategy consists of two basic steps:

1. The target profile, $t$ is compared to each profile case, $s \epsilon S$, to select the $k$ most similar cases.
2. The items contained within these selected cases (but absent in the target profile) are ranked according to the relevance to the target, and the $r$ most similar items are returned as recommendations.

## Profile Matching

The profile similarity metric is presented in Equation 4 as the weighted-sum of the similarities between the items in the target and source profile cases. In the situation where there is a direct correspondence between an item in the source, $s_i$, and the target, $t'_j$, then maximal similarity is assumed (Equation 5). However, the nature of ratings-based profile cases is such that these direct correspondences are rare and in such situations the similarity value of the source profile item is computed as the mean similarity between this item and the $n$ most similar items in the target profile case ($t'_1 ... t'_n$) (Equation 6).

$$PSim(t, s, n) = \sum_{s_i \epsilon s} w_i \cdot ISim(t, s_i, n) \tag{4}$$

$$ISim(t, s_i, n) \quad = \quad 1 \; if \; t'_j = s_i \tag{5}$$

$$= \quad \frac{\sum_{j=1..n} sim(t'_j, s_i)}{n} \tag{6}$$

Notice, that if $n = 1$ and there is a perfect one-to-one correspondence between the target and source profile cases, then this profile similarity metric is equivalent to the traditional weighted-sum similarity metric.

## Recommendation Ranking

Once the $k$ most similar profile cases ($\overline{S}$) to the target have been identified, a set of ranked item recommendations can be produced. There are three factors to consider when ranking these recommendations. First, we want to give priority to those items that have a high similarity to the target profile case. Second, items that occur in many of the retrieved profile cases should be preferred to those that occur in few profile cases. Finally, items recommended by profiles similar to the target should be preferred to items recommended by less similar profiles. Accordingly we compute the *relevance* of an item, $s_i$ from a retrieved profile case, $s$, with respect to the target profile, $t$, as shown in Equation 7;

where $S' \subseteq \overline{S}$ is the set of retrieved profile cases that contain $s_i$.

$$Rel(s_i, t, \overline{S}) = ISim(s_i, t, k) \cdot \frac{|S'|}{|\overline{S}|} \cdot \sum_{s \epsilon S'} PSim(s, t) \quad (7)$$

## Experimental Evaluation

In order to evaluate our approach to mining and applying similarity knowledge, we conducted a series of experiments using data from 622 PTV customer profiles. The first set of experiments were designed to investigate the performance characteristics of our chosen data mining algorithm within the PTV domain. The second set of experiments tested the potential for mining additional similarity knowledge, in terms of relationships between programme items. The third set of experiments tested the potential of the approach for improving actual recommendation quality.

We employ modified versions of PTV's user profiles as rating cases. Each profile case contains a list of programmes that the user has previously rated as positive; for now we will ignore the negative ratings and also the rating values themselves, leaving these factors for future work.

### Tuning Data Mining

In our first set of experiments, we applied the Apriori algorithm to our PTV data set for different parameterizations of the algorithm. Since the data mining was the basis for maintaining similarity knowledge, we wanted to determine how rule generation would be influenced by parameter choice, namely confidence and support thresholds. The first experiment tested the number of rules generated for varying levels of confidence and support. Changes in the number of rules across confidence values for different support levels were quite similar, indicating that the parameterization is more dependent on confidence than on support. This can be seen more clearly in the results of our second experiment, shown in Figure 1; there is little change in rule accuracy as the level of support changes across different levels of confidence. Average accuracy here measures how well the generated rules match the entire profile set, and is computed as the ratio of antecedent and consequent matches to antecedent matches. Based on these results, we chose a representative support level (5%) for the remainder of the experiments.

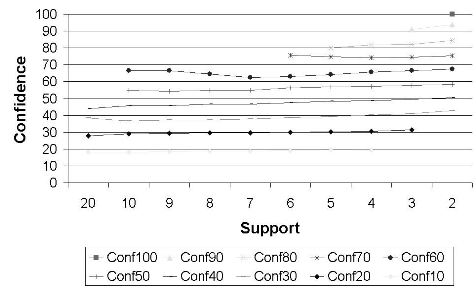


Figure 1: Average accuracy as a factor of confidence and support.

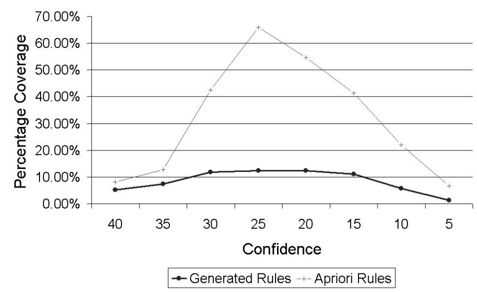


Figure 2: Programme matrix coverage.

### Increasing Coverage

In the next set of experiments, we were interested in evaluating the degree to which similarity coverage could be improved by the rule sets. For the first experiment, the density of the generated programme-programme similarity matrix was taken as our measure of similarity coverage. We varied confidence levels from 40% to 5% at 5% intervals. On each run we generated the Apriori direct rule set, as well as a maximal indirect rule set and filled in the programme-similarity matrix, taking the matrix density relative to the programmes that participated in the rule set. The results are shown in Figure 2. The direct item similarities provide an average of 10% coverage, but there is a marked increase for the indirect similarity rules. We note a maximum of approximately 65% coverage in the best case.

Of course the Apriori method does not generate association rules for every profile programme, and in fact a great many programmes are ignored by Apriori because their frequency fails the Apriori thresholds. Nevertheless, when we add these newly generated rules to the collaborative filtering matrix, which is a programme-programme similarity matrix across all programmes, we are able to increase its starting density from 0.6% to 2.6%.

We also wanted to test the potential for exploiting rule symmetry. Again, we varied the confidence and computed a rule symmetry correlation coefficient between reversed rules, $A \Rightarrow B$, $B \Rightarrow A$ for depths of direct/indirect rule generation from 1 to 5. As confidence decreases, indirect rules improve the correlation, but the correlation remains fairly low (within .3 for most levels). For high levels of confidence (35 and 40), there was a strong correlation (above .8) and indication that symmetry would hold, but this is balanced by the reduction in overall programme matrix coverage at that level of confidence. Thus for the following experiments, we have not made use of symmetry in building the programme similarity matrix, instead noting it for future work.

## Improving Recommendations

With encouraging results for increasing similarity coverage, we designed experiments to test the effect of this new similarity knowledge on recommendation quality. For these experiments, we chose a representative confidence level of 10%. In the first experiment, we measured the number of profiles that could possibly be compared as a percentage of all potential profile comparisons for different similarity metrics. Using our standard collaborative filtering metric gave a coverage of 42%, while our similarity metric using direct rules only and then adding indirect rules increased the coverage to 52% and 70% respectively.

The final experiment tested the accuracy of recommendation. After generating direct and indirect similarity rules, a profile case is selected from the case-base. A parameterized percentage of the programme items in the selected case are removed from consideration. The remainder of the profile case is then used as the basis for retrieval, using our similarity metric and recommendation ranking described previously. Recommender accuracy was calculated as the number of removed target case items found in the recommendation set. We were most interested in looking for an uplift in recommendation quality when comparing our CBR technique (using direct and indirect similarity rules) to the pure collaborative filtering technique. When using the similarity knowledge generated from the direct Apriori rules we find an average quality uplift of 93% over varied levels of item removal; that is our recommendation technique is generating almost twice as many successful recommendations as the pure collaborative filtering method.

When we included the indirect similarity knowledge, there was also a quality uplift, but this time it is only an average of 32%. On the face of it, then, it seems that the indirect similarity rules must not be as accurate as the direct rules. However another possibility is that these new rules are capable of generating high quality recommendations but recommendations that are simply not present in the limited set of target programmes that were blocked out from the profile cases. Our future work will endeavour to investigate this issue in more detail.

## Conclusions

We have described a case-based reasoning perspective on collaborative filtering that employs ratings-based profiles directly as cases and that responds to the similarity coverage problem by mining and applying new similarity knowledge. A preliminary evaluation of our work has been conducted in the domain of TV listing recommendation, using a well known commercial recommender (PTV - (Smyth & Cotter 2001)). The results demonstrate the effect of using collaborative filtering profiles as cases in a case-based recommender, and the improvement of overall recommendation quality by learning similarity knowledge using profile mining. Through testing on further datasets, using multiple antecedents and consequents in data mining, inclusion of negative ratings and comparison to existing recommender algorithms, we hope to increase the effectiveness of our approach in future work.

## References

Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; and Verkamo, A. I. 1995. Fast discovery of association rules. In Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; and Uthurusamy, R., eds., *Advances in Knowledge Discovery and Data Mining*. AAAI Press. 307–328.

Aha, D.; Breslow, L.; and Muñoz-Avila, H. 2001. Conversational case-based reasoning. *Applied Intelligence* 14(1):9–32.

Burke, R. 2000. A case-based reasoning approach to collaborative filtering. In *Proceedings of EWCBR-00*, 370–379. Springer.

Burkhard, H.-D. 1998. Extending some concepts of CBR-foundations of case retrieval nets. In Lenz, M.; Bartsch-Spörl, B.; Burkhard, H.-D.; and Wess, S., eds., *Case-Based Reasoning Technology: From Foundations to Applications*. Springer. chapter 2.

Doyle, M., and Cunningham, P. 2000. A dynamic approach to reducing dialog in on-line decision guides. In *Proceedings of EWCBR-00*, 49–60. Springer.

Hayes, C.; Cunningham, P.; and Smyth, B. 2001. A case-based view of automated collaborative filtering. In *Proceedings of ICCBR-01*, 234–248. Springer.

Hipp, J., and Nakhaeizadeh, U. G. 2000. Mining association rules: Deriving a superior algorithm by analyzing today's approaches. In *Proceedings of the 4th European Symposium on Principles of Data Mining and Knowledge Discovery*.

Leake, D. B., and Wilson, D. C. 2001. A case-based framework for interactive capture and resuse of design knowledge. *Applied Intelligence* 14(1).

Lenz, M.; Auriol, E.; and Manago, M. 1998. Diagnosis and decision support. In Lenz, M.; Bartsch-Spörl, B.; Burkhard, H.-D.; and Wess, S., eds., *Case-Based Reasoning Technology: From Foundations to Applications*. Springer. chapter 3.

Minor, M., and Hanft, A. 2000. Corporate knowledge editing with a life cycle model. In *Proceedings of GWCBR-00*.

Smyth, B., and Cotter, P. 2001. Personalized electronic programme guides. *Artificial Intelligence Magazine* 21(2).

Smyth, B., and McKenna, E. 1998. A portrait of case competence: Modelling the competence of case-based reasoning systems. In *Proceedings of EWCBR-98*, 208–220. Springer.

Wilson, D. C., and Leake, D. B. 2001. Maintaining case-based reasoners: Dimensions and directions. *Computational Intelligence* 17(2).