Preserving Recommender Accuracy and Diversity in Sparse Datasets *

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

Abstract

Recent research has shown that a case-based perspective on collaborative filtering for recommendation can provide significant benefits in decision support accuracy over traditional collaborative techniques, particularly as dataset sparsity increases. These benefits derive both from the use of more sophisticated case-based similarity metrics and from the proactive maintenance of item similarity knowledge using data mining. This paper presents a natural next step in the work by validating these findings in the context of more complex models of collaborative filtering, as well as by demonstrating that such techniques also preserve recommendation diversity.

Introduction

The Internet and its associated e-commerce services have proven a fertile ground for the development of a new breed of artificial intelligence technologies and applications. For example, the field of recommender systems is motivated by the need for improved search capabilities that help users and customers to more efficiently locate information items that correspond to their needs and preferences (Resnick & Varian 1997). Recommender systems combine ideas from user modelling, machine learning, information filtering and user-interface design in order to provide personalized information retrieval services on a user by user basis.

In particular, one of the novel techniques that has developed out of work in this field is *automated collaborative filtering* (ACF), which leverages the usage history of groups of similar users in order to make recommendations to a target user (Konstan *et al.* 1997; Smyth & Cotter 2001; Terveen *et al.* 1997). ACF has proven particularly successful in domains where background knowledge and rich content descriptions of the recommendable items are not available, rendering more traditional content-based techniques useless. For example, ACF techniques have been used to drive high-quality movie recommendations in systems such as MovieLens (Resnick *et al.* 1994) and EachMovie (McJones 1997) even though no information is available about the genre or other useful details of any individual movie—the satisfaction ratings of users alone drive the collaborative filtering process.

The obvious success of collaborative filtering aside, there are a number of problems with this technique. In particular, its success—in terms of its ability to make accurate item recommendations for a target user—is critically dependent on the sparsity (or density) of the profile space. For example, in many collaborative filtering systems there are many potentially recommendable items and many user profiles, but a typical user may have rated only a tiny percentage of these items. As a result, the degree of overlap between individual users—which has a significant impact on the ability of ACF to recognise similarities between users—is likely to be small. Thus the user-item ratings matrix is sparsely populated. In practice, this will often make it difficult to find a sufficiently diverse group of recommendation partners that are similar enough to the target user to act as a reliable source of recommendations. Solving the sparsity problem will have a significant impact on the general applicability and overall performance of collaborative recommenders.

In our research we are concerned with investigating and solving this sparsity problem by taking a case-based perspective on collaborative filtering. In previous work (O'Sullivan, Wilson, & Smyth 2002b) we described a case-based approach to collaborative recommendation that employs data-mining techniques to maintain similarity knowledge by proactively discovering item relationships in ratings profiles. Moreover, we have shown how this similarity knowledge can drive a content-based recommendation approach to deliver high quality recommendations that are superior to those generated by a traditional collaborative filtering model using the same ratings profiles. In addition, we have argued that our approach has significant benefits when it comes to generating recommendations in sparse data sets. Specifically, we have demonstrated a strong correlation between the accuracy benefit of our approach, relative to collaborative filtering, and the sparsity of the data set (O'Sullivan, Wilson, & Smyth 2002a). In short, sparse data sets allow us to deliver significant benefits over traditional collaborative filtering, benefits that reduce gracefully as data set density increases.

Since our experiments in previous work compared against a traditional model of collaborative filtering—essentially a straightforward nearest-neighbour method based on a mea-

*The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

sure of direct profile overlap—a natural next step in the research is to validate our findings in comparison to more recent and sophisticated models of collaborative filtering. Moreover, because the traditional collaborative filtering approach of recommending other users (and thereby items) fosters an inherent diversity in recommendation, we are interested in analysing the impact of our approach on recommendation diversity.

In this paper we describe recent work that compares our case-based approach to collaborative recommendation techniques that employ user-based cosine (Sarwar *et al.* 2000) and item-based (Sarwar *et al.* 2001) metrics. Once again, we demonstrate the benefits offered by our recommendation approach on a variety of different data sets. And once again, we notice a strong correlation between these benefits and the sparsity of the data sets. Like earlier collaborative results, user-based cosine and item-based techniques are challenged in sparse data sets, whereas our approach achieves significantly higher prediction accuracies. We also show that our approach preserves a high degree of recommendation diversity with respect to traditional collaborative techniques.

Recommendation Strategies

In general, we were interested in comparing the decision support accuracy of our approach with more sophisticated ACF approaches. In particular, we selected the collaborative methods (user-based cosine similarity, as well as item-based cosine and probability similarity) implemented in the freely available SUGGEST recommendation engine (Karypis 2000) for comparison. This also provided the benefit of comparing against an independent implementation of the techniques. In order to set the stage for comparing recommendation approaches, we give a brief overview of the strategies involved.

User-Based Strategies

User-based strategies first employ a measure of similarity between user profiles in order to select a set of maximally similar neighbours to a given target user. They go on to combine the preferences of those neighbour users in order to rank and select a set of items for recommendation. The user-based strategies we compare are a traditional direct-overlap similarity, as well as the SUGGEST cosine similarity method, detailed in (Karypis 2000; Sarwar *et al.* 2000).

User Overlap Similarity: As our baseline for comparison, we employ a simple user-based ACF technique that measures the similarity between two users a and b by finding the percentage of direct overlap, as shown in equation 1.

$$Usim(a,b) = \frac{a \cup b}{min(\|a\|, \|b\|)} \tag{1}$$

The items that occur in the k closest neighbour profiles, but not in the target profile, are combined by selecting the N most frequently occurring items, and these top-N items are recommended to the user.

User Cosine Similarity: With cosine similarity, two users a and b are viewed as vectors in the space of items. The similarity between them is measured by calculating the cosine of the angle between these two vectors, using the formula:

$$Usim(a,b) = cos(\overrightarrow{a}, \overrightarrow{b}) = \frac{\overrightarrow{a} \cdot \overrightarrow{b}}{\|\|\overrightarrow{a}\|\|_2 * \|\|\overrightarrow{b}\|\|_2} \tag{2}$$

Again, the k most similar neighbour user profiles are found using this measure, and the top-N frequently occurring items that are not already in the target profile are returned as recommendations.

Item-Based Strategies

Item-based strategies first define a measure of similarity between items that could appear in a user's profile. This item similarity metric is then used to define a measure for the relevance of a particular item to an entire target user profile, which allows items to be ranked and selected for recommendation. The item-based strategies we compare are the SUGGEST cosine and probability similarity methods, detailed in (Karypis 2000; 2001).

Item Cosine Similarity: For this method each item is treated as a vector in the space of users and the cosine measure between these vectors is used to measure similarity. Thus for an n x m user-item matrix, the similarity between two items v and u can be defined as the cosine of the n dimensional vectors corresponding to the v^{th} and u^{th} matrix column (as in Equation 3).

$$Isim(v,u) = cos(\overrightarrow{v}, \overrightarrow{u}) = \frac{\overrightarrow{v} \cdot \overrightarrow{u}}{\|\|\overrightarrow{v}\|\|_2 * \|\|\overrightarrow{u}\|\|_2} \tag{3}$$

The similarity between a target user's set of profile items U and an item $v \notin U$ is calculated by summing the item similarities between v and each item in U. In order to account for variations in similarity, such as for overlaps in infrequently occurring items, the similarities for each item are normalised to a unit sum. The items are sorted by similarity score, and the top-N most similar items are returned as the recommended set.

Item Probability Similarity: Another way of computing the similarity between two items v and u is based on the conditional probability of selecting one item (u) given that the other item (v) has already been selected by a user, denoted $P(u|v)$. This conditional probability can be found by calculating the number of users that prefer both items v and u divided by the total number of users that prefer u,

$$Isim(v,u) = P(u|v) = \frac{Freq(uv)}{Freq(v)} \tag{4}$$

where $Freq(X)$ is the number of users that favour the items belonging to the set X. This can also be scaled by a parameter α to allow for biased similarities when some of the items occur very frequently:

$$Isim(v,u) = \frac{Freq(uv)}{Freq(v) * (Freq(u))^{\alpha}} \tag{5}$$

As with the item cosine measure, the item-to-target-profile similarities are computed and normalised, and the top-N most similar items are recommended.

Case-Based Strategy

Our case-based approach addresses the sparsity problem by first applying data mining techniques, specifically association rule mining, to a set of ratings-based user profile data in order to derive indirect similarity knowledge composed as rules that relate items. These association rules are of the form $A \Rightarrow B$, where A and B are sets of items. In data mining terms, whenever a transaction (case) contains a certain itemset 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. These confidence values are taken as probabilities and used to fill in an item-item similarity matrix, which provides the additional similarity knowledge necessary to compare non-identical profile items (O'Sullivan, Wilson, & Smyth 2002b).

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

1. The target profile, t is compared to each profile case, $c \epsilon C$, 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 relevant items are returned as recommendations.

Profile Matching: The profile similarity metric (Equation 6) is computed as the weighted-sum of the similarities between items in the target and source profile cases. In the situation where there is a direct correspondence between an item in the source, c_i, and the target, t_j, then maximal similarity is assumed (Equation 7). 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 8).

$$PSim(t, c, n) = \sum_{c_i \epsilon c} w_i \cdot ISim(t, c_i, n) \tag{6}$$

$$\begin{aligned} ISim(t, c_i, n) &= 1 \ if \ \exists \ t_j = c_i & (7) \\ &= \frac{\sum_{j=1..n} sim(t_j, c_i)}{n} & (8) \end{aligned}$$

Recommendation Ranking: Once the k most similar profile cases ($\hat{C}$) to the target have been identified, their items are combined and ranked for recommendation using three factors. We prioritise (1) items that have a high similarity to the target profile case, (2) items that occur in many of the retrieved profile cases, and (3) items that are recommended by profiles most similar to the target. Accordingly we compute the *relevance* of an item, c_i, from a retrieved profile case, *c*, with respect to the target profile, *t*, as shown in Equation 9; where $C' \subseteq \hat{C}$ is the set of retrieved profile cases that contain c_i.

$$Rel(c_i, t, \hat{C}) = ISim(c_i, t, k) \cdot \frac{|C'|}{|\hat{C}|} \cdot \sum_{c \epsilon C'} PSim(c, t, k) \tag{9}$$

Finally, the top-N ranked items are returned for recommendation.

Experiments

In order to evaluate these approaches to recommendation, we tested all five strategies. Previous experiments have discussed parameterization for our case-based technique (O'Sullivan, Wilson, & Smyth 2002b), and the only parameter needing tuning in the SUGGEST system, α, is automatically calculated for each dataset.

Datasets

We conducted experiments using 4 datasets in the television and movie domains:

1. PTVPlus dataset consisting of 622 user profiles;
2. Físchlár dataset consisting of 650 user profiles;
3. MovieLens dataset consisting of 659 user profiles;
4. EachMovie dataset consisting of 651 user profiles;

Each dataset used for these experiments is a random selection of profiles from larger datasets.

Algorithms

We use 5 different algorithms in testing the aforementioned recommender strategies:

1. Direct - our case-based approach;
2. CF - simple user-based direct overlap
3. SUGGEST Item Cos - item-based cosine similarity;
4. SUGGEST Item Prob - item-based probabilistic similarity;
5. SUGGEST User Cos - user-based cosine similarity.

Method

To evaluate our techniques, we split each dataset into training (30%) and test (70%) subsets. The training subset is used for model generation (not required in CF) and the test subset is used in querying recommendations. In testing, we take each profile and block out a parameterised percentage of the profiles items; the unblocked items are used to generate recommendations which are then tested against the blocked out items.

We use two metrics in calculating accuracy:

Metric 1: Percentage of removed items that were recommended in each case.

Metric 2: Percentage of profiles in which at least one removed item was recommended.

Metric 1 is a stringent accuracy criterion, and a result of 100% here means the system is able to recommend all of the blocked out items. Metric 2 serves as a looser measure of accuracy where we focus on the ability of a system to generate at least one useful recommendation.

Results

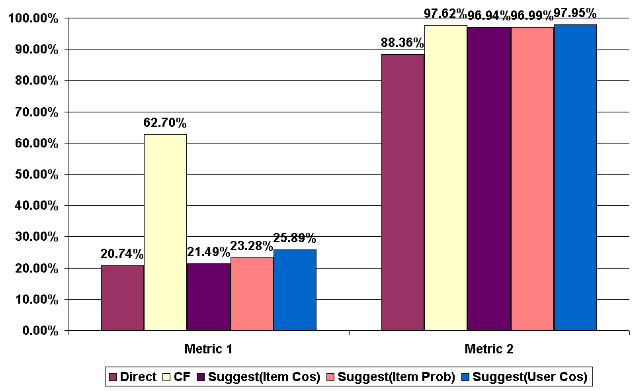

Figure 1: MovieLens Recommendation Accuracy

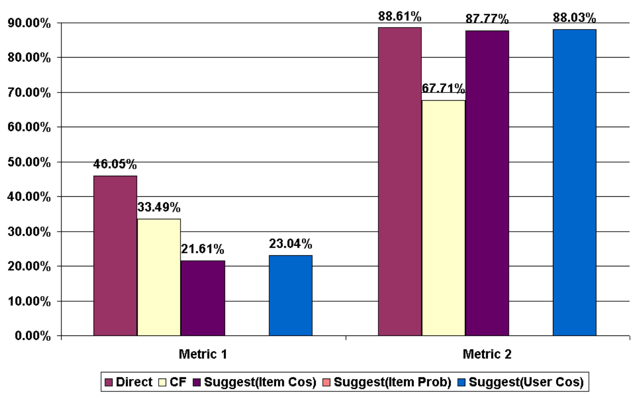

Figure 2: EachMovie Recommendation Accuracy

We can see from the accuracy result graphs (Figures 1-4) that the MovieLens dataset has comparable accuracy results for both CF and SUGGEST models with our (Direct) case-based approach performing less impressively; for example, Direct achieves an accuracy level (Metric 2) of approximately 88% compared to accuracies of greater than 96% for the competing methods. The fact that CF outperforms all models in both metrics suggest a inherent bias that is probably due to the high density of the MovieLens dataset (Figure 1). In the other datasets (Figures 2, 3, 4), our technique outperforms both CF and SUGGEST in all cases. And while the SUGGEST methods perform comparably with Direct in EachMovie but each fall in density causes a decrease in accuracy.

To elaborate for a moment on this important density issue, we can implement the density metric shown in Equation 10

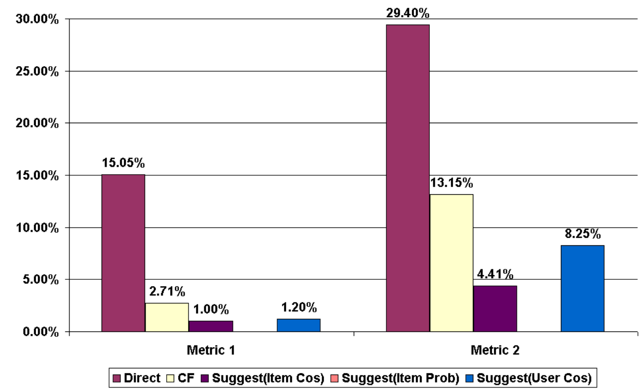

Figure 3: Físchlár Recommendation Accuracy

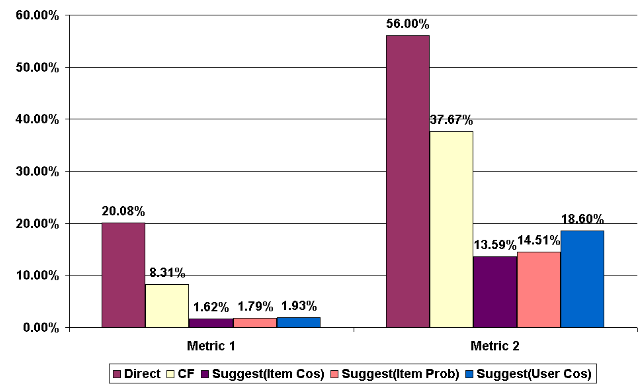

Figure 4: PTVPlus Recommendation Accuracy

which is adapted from (Sarwar *et al.* 2001):

$$Density(Dataset) = \frac{\#\ of\ nonzero\ entries}{Total\ \#\ of\ entries} \qquad (10)$$

where the number of total entries is calculated by multiplying the number of users by the number of items that have been rated at least once; the number of nonzero entries is the total number of ratings overall in the dataset. Table 1 shows the results of this metric on the 4 datasets (with Físchlár as the baseline):

Dataset	Density	Percentage Increase
Físchlár	0.00358	-
PTVPlus	0.00575	60%
EachMovie	0.03614	910%
MovieLens	0.06603	1744%

Table 1: Dataset Density

We conclude that our case-based approach has the ability to generate intelligent recommendations in sparse datasets, an area where other algorithms degrade much less gracefully. This is summarized in Figure 5, which shows the increase in accuracy (left axis) relative to the baseline (overlap) ACF results as density (right axis) increases.

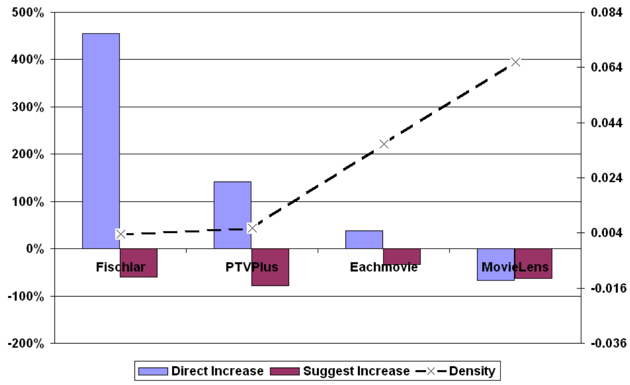

Figure 5: Metric 1 vs Density

We have also been able to show from the two movie datasets (MovieLens & EachMovie) that diversity is preserved (within 3% of the maximum (CF) value), even though we have higher recommender accuracy and increased sparsity pressures. Diversity is calculated by performing simple overlap among the genre listings available with each dataset.

In the Físchlár and EachMovie datasets, we were unable to run the SUGGEST tests for the item-based probability strategy. We suspect this is due to some constraints that exist either in construction of the dataset, or in the way in which the SUGGEST model is generated as we were unable to evaluate either the α parameter or initialise the model with this parameter manually.

Conclusion

In this paper we set out to build on previous work which investigated the use of association-rule mining and case-based methods in recommender systems. The earlier work showed how our approach was capable of generating more accurate recommendations than a traditional user-based collaborative approach. Interestingly, we also noticed a strong relationship between the benefits of this new recommender and the sparsity of the underlying dataset. Although the traditional collaborative approach was found to be significantly hampered by sparse datasets, our technique succeeded in maintaining respectable levels of recommendation accuracy.

In the current paper we have extended this research to cover more sophisticated and recent user- and item-based collaborative recommendation methods. The results are consistent. Our case-based recommender is capable of achieving significant accuracy wins, and the significance of these wins is enhanced in sparse datasets.

We conclude, therefore, that at a fundamental level, the combination of association rule mining and case-based methods offers recommender systems some considerable protection against accuracy problems normally associated with the sparsity issue. Competing collaborative techniques suffer when faced with sparse datasets, thereby limiting recommendation accuracy. The same is not true of our new case-based method which achieves significantly higher accuracy while maintaining the diversity benefits of collaborative approaches.

References

Karypis, G. 2000. Suggest Top-N Recommendation Engine, http://www-users.cs.umn.edu/ karypis/.

Karypis, G. 2001. Evaluation of Item-Based Top-N Recommendation Algorithms. In *CIKM*, 247–254.

Konstan, J.; Miller, B.; Maltz, D.; Herlocker, J.; Gordon, L.; and Riedl, J. 1997. Grouplens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM* 40(3):77–87.

McJones, P. 1997. Eachmovie Collaborative Filtering Dataset, DEC Systems Research Center, http://www.research.compaq.com/src/eachmovie/.

O'Sullivan, D.; Wilson, D.; and Smyth, B. 2002a. Improving Case-Based Recommendation: A Collaborative Filtering Approach. In Craw, S., and Preece, A., eds., *Proceedings of the 6th European Conference on Case Based Reasoning (ECCBR-02)*, 278–291.

O'Sullivan, D.; Wilson, D.; and Smyth, B. 2002b. Using Collaborative Filtering Data in Case-Based Recommendation. In *Proceedings of the 15th International FLAIRS Conference*.

Resnick, P., and Varian, H. R. 1997. Recommender Systems. *Special Section in Communications of the ACM* 40(3).

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; and Riedl, J. 1994. Grouplens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work*, 175–186.

Sarwar, B. M.; Karypis, G.; Konstan, J. A.; and Riedl, J. 2000. Analysis of Recommendation Algorithms for E-commerce. In *ACM Conference on Electronic Commerce*, 158–167.

Sarwar, B. M.; Karypis, G.; Konstan, J. A.; and Riedl, J. 2001. Item-based Collaborative Filtering Recommender Algorithms. In *Proceedings of the Tenth International World Wide Web Conference*.

Smyth, B., and Cotter, P. 2001. Personalized Electronic Programme Guides. *Artificial Intelligence Magazine* 21(2).

Terveen, L.; Hill, W.; Amento, B.; McDonald, D.; and Creter, J. 1997. PHOAKS: A System for Sharing Recommendations. *Communications of the ACM* 40(3):59–62.