

Sticking with a Winning Team: Better Neighbour Selection for Conversational Collaborative Recommendation^{*}

Rachael Rafter, Lorcan Coyle, Paddy Nixon, and Barry Smyth

Adaptive Information Cluster,
School of Computer Science and Informatics,
UCD Dublin, Ireland
`firstname.lastname@ucd.ie`

Abstract. Conversational recommender systems have recently emerged as useful alternative strategies to their single-shot counterpart, especially given their ability to expose a user’s *current* preferences. These systems use conversational feedback to hone in on the most suitable item for recommendation by improving the mechanism that finds useful collaborators. We propose a novel architecture for performing recommendation that incorporates information about the individual performance of neighbours during a recommendation session, into the neighbour retrieval mechanism. We present our architecture and a set of preliminary evaluation results that suggest there is some merit to our approach. We examine these results and discuss what they mean for future research.

1 Introduction

Traditionally, collaborative recommender systems are based on a *single-shot* model of recommendation where a single set of recommendations is generated based on a user’s (past) stored preferences [1]. Such systems assume that users have stable preferences which can be represented with a static user profile that grows over time. However, content-based recommender system research has begun to look towards more *conversational* models of recommendation, where the user is actively engaged in directing search at recommendation time [2–4].

Previously we have proposed adopting a similar model for collaborative recommendation [5] where recommendations are made in a cyclical process, and the user provides *preference-based feedback* [6] on the suitability of the items recommended after each *cycle*. This feedback is used to model the user’s current *short-term* preferences. This has been motivated by our belief that a user’s *current* preferences are influenced by her current mood, and moreover that they may deviate from her regular movie preferences. Hayes and Cunningham [7] have similarly pointed out that Collaborative Recommendation can lack sensitivity to a user’s current interests and may cause frustration and distrust. They

^{*} This work is partially supported by Science Foundation Ireland under grant number 04/RPI/1544 “Secure and predictable pervasive computing”.

propose to boost collaborative recommendations with a more knowledge-heavy approach than ours, that incorporates context or task driven information into the recommendation process. Although it could be argued that this conversational approach imposes a burden on the user to provide feedback, we would maintain that it can also help the user to explore the information space in a more useful manner.

Recent results from our research [5] have already indicated that this conversational model of recommendation has advantages over its static single-shot counterpart. Bridge and Kelly [8], went further and incorporated diversity into the recommendation process which improved results. In this paper, we focus on enhancing the selection of *nearest neighbours* (NNs) which is crucial to any recommender system. We propose that the feedback provided by the user during a session, can not only be used to model the user’s current preferences, but also to identify those neighbours that are performing best as recommendation partners for the target user, given her current preferences.

Our contributions in this article are twofold. We propose a novel component called the *Sticky Layer* that controls the promotion and demotion of neighbours based on their performance to date during a recommendation session; this layer performs a type of “recommendation priming” that is guided by the recommendation session. We also propose an extension of Hayes *et al.*’s [9] Case Retrieval Net (CRN) implementation of k-NN retrieval for Automated Collaborative Filtering, which takes the effectiveness of an individual user’s past recommendations in the current recommendation session into account, when retrieving new recommendations. The CRN is supported by the Sticky Layer.

The rest of this paper is organised as follows: In Section 2 we provide some background. In Section 3 we discuss the novel aspects of our system. In Section 4 we discuss some preliminary evaluation results, and in Section 5 we conclude and propose future research. We should note here that the work presented in this article is still not fully mature, and our evaluation reflects ongoing work. However, these preliminary results present some interesting ideas that have emerged from a considerable amount of research. We try to raise some open questions, which we hope will stimulate interesting future research.

2 Background

In this section we discuss collaborative recommendation and conversational recommendation in brief, and then we provide some background on Case Retrieval Nets and how they can be used for collaborative recommendation.

2.1 Collaborative Recommendation

Single-shot Collaborative Recommendation (SS-CR) is a *content-free* recommendation strategy based on the premise that similar users like similar items. Recommendations are compiled for the *target user* based on the profiles of her *nearest neighbours* (NNs). Importantly, the profile only contains information about the

user’s *long-term* preferences, and ignores any *short-term* preference differences that the user may have. Therefore user profiles where *both* short-term and long-term preferences of the user are represented, have been proposed, e.g. [10, 11].

In Conversational Collaborative Recommendation (C-CR) [5] the standard long-term profile from the single-shot model, is complemented by a short-term profile which models the user’s more current (and possibly transient) preferences. Here, *cycles* of k item recommendations are made to the user; after each cycle she is asked to indicate which recommendation would be most suitable, or else indicate that none are suitable (by selecting one recommendation, or by rejecting all of them, respectively). This feedback is then incorporated into her short-term profile and used to alter the similarity retrieval process. This process is repeated, with new items being recommended to the user each time based on the updated profile, until the user finds an item with which she is satisfied. We refer to the entire set of cycles resulting in a satisfactory recommendation as a *session*.

2.2 CRNs for Collaborative Recommendation

Hayes *et al.* [9] pointed out the similarities between Collaborative Recommendation and Case-Based Reasoning (CBR). Core to the CBR methodology is the use of a similarity retrieval mechanism for finding the k nearest neighbours (k-NN) to the current problem (or target) case. Case Retrieval Nets (CRNs) have been used to implement k-NN similarity retrieval due to their ability to efficiently and flexibly retrieve similar cases [12]. CRNs outperform the standard k-NN approach in domains where there is feature-value redundancy or domains with many missing feature-values.

Hayes *et al.* proposed that CRNs could be used to perform collaborative recommendation (or Automated Collaborative Filtering) by treating a user profile like a case and the user recommendations on individual items as case features [9]. These cases are usually sparse, playing to the strengths of the CRN, and allowing for efficient retrieval of the most similar users to the current target case. Information from the NNs that emerge can then be used to make a recommendation to the target user. This approach had two major limitations:

- Features are represented as item recommendation pairs (IEs), whereby the CRN typically contains more than one IE for each item, (one IE for every distinct recommendation score for that item).
- Similarity measures need to be explicitly defined between IEs. This typically means that a domain expert must specify how any two features relate to each other. This implies a knowledge-intensive approach.

To overcome these limitations, we take inspiration from the work of Delany *et al.* [13] on using CBR in the spam filtering domain. They used CRNs for retrieving similar e-mails to classify a target e-mail. The spam-filtering domain is interesting for CRN development since all features are binary — spam features just represent words in an e-mail, each feature is true if the word it represents is present in an email, and false otherwise. With binary features, the CRN can

be connected in such a way that calculating similarity is not necessary (the assumption is that similarities between features are always zero).

Given that the use of the CRN structure comes from the CBR field it should be pointed out that the process of incorporating user feedback into a similarity retrieval process as we do has an analogue there. Richter [14] has proposed that CBR has four distinct knowledge containers (the case-base, vocabulary used, similarity measure, and solution transformation) and that it may be advantageous to move knowledge between them. The CBR analogue to our work would be a system that moved (or incorporated) knowledge from the solution transformation to the similarity measure.

3 Implementation

Recommendations are generated using user profiles retrieved from a novel representation of a CRN called a Collaborative Case Retrieval Net (*CCRN*). Our CCRNs are an adaptation of the CRN similarity retrieval implementation from the Fionn CBR framework [15].

Traditional CRNs are made up of the following components (the differences between CRNs and CCRNs are also outlined), which are illustrated in Fig. 1:

- Nodes represent stored CBR cases (traditional CRNs call these *case nodes*). CCRN nodes represent individual user profiles.
- Traditionally, Information Entities (IEs) represent feature-value pairs within cases. CCRN IEs represent items that are available for recommendation.
- Relevance Arcs link case nodes with the IEs that represent them. Typically they have weights that capture the importance of the IE to the connecting node. CCRNs use these to capture both the items that were preferred by the user as part of their permanent profile and items that were preferred by a target user during a recommendation session.
- Similarity Arcs connect IEs that refer to the same features, and have weights relative to the similarity between connected IEs. CCRNs makes the assumption that similarities between IEs are always zero, and so these are not included. It should be noted that in Figure 1 only six similarity arcs are shown - a typical CCRN should contain $(n - 1)!$ arcs for every n IEs.

When performing collaborative recommendation, the user who is seeking a recommendation is presented to the CCRN as a target node (T in Figure 1). User T has previously liked item 2, which was also liked by User a so those lines have arrows reflecting permanent activation; *in the current recommendation cycle* User T has liked item 4, which was also liked by users a and c , so those relevance arcs are darker and have arrows to reflect temporary activation. As the recommendation session continues, further relevance arcs will be created to express new preferences. Activation is spread across the net structure (shown by the arrowed relevance arcs) and accumulated at the neighbouring nodes. In the example diagram, neighbour a would be returned as the target user's NN with an activation of 2 (c would be in second place with an activation of 1).

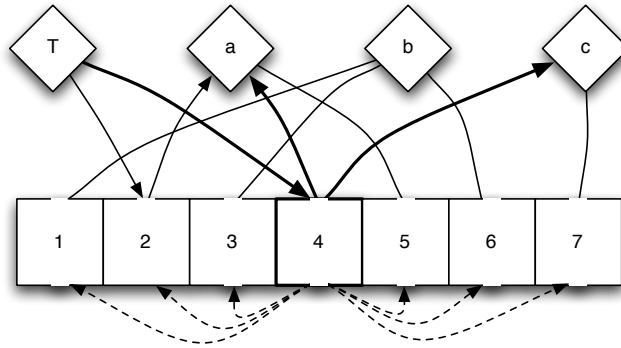


Fig. 1. CCRN Architecture. The diamonds at the top are nodes representing users, where T is the target user. Squares 1 to 7 are IEs representing 7 items that are available for recommendation. The lines connecting IEs to Nodes are relevance arcs, relating items to users that liked them in the past; the broken arrows below the IEs represent similarity arcs from IE 4 to all other IEs in the net, which only exist in traditional CRNs and are shown here only for comparative illustration.

This architecture provides behaviour that is no different from typical collaborative recommendation approaches. However, because of the nature of the CRN structure, it is possible to inject additional information into the retrieval process. We inject information about useful users for recommendations into the “Sticky Layer” of the CCRN.

3.1 The Sticky Layer

Our hypothesis is that in order to maintain a consistent recommendation process towards a stated goal, it is best to reuse recommendations from users that have proven to be responsible for good recommendations during a session so far. The analogy would be that the user is taking advice from a subset of the users that are providing recommendations and following these trusted users’ advice over a number of cycles. Thus, the CCRN incorporates a measure of context at each cycle in the recommendation process.

We use a feedback mechanism called the *Sticky Layer*, which makes good neighbours *stick*, so they are more likely to be re-chosen as recommendation partners for the target user during a given recommendation session. This layer adds a memory to the recommendation session and performs a priming function with respect to neighbour selection. The job of the Sticky Layer is to promote or demote neighbours based on their performance as recommendation partners to date in a session. At the end of each cycle, every NN to the target user in that cycle, receives a *sticky boost*, (positive or negative), depending on the quality of the neighbour’s *contribution* to the recommendations in the cycle. Sticky boosts are currently in units of 1. The sticky boost can be updated in three ways:

- if the target user selects one of the recommended items as preferable in a cycle and a NN has *contributed* to that item being recommended (that item is in her profile) we increment her sticky boost (by 1).
- Similarly, we can decrement the sticky boost of a NN if she did not contribute to that item being recommended.
- We can also decrement the sticky boost of a NN if she contributed to a set of recommendations made in a cycle in which all of the (bad) recommendations were rejected by the target user.

This boost is incorporated at the Sticky Layer of the CCRN, where it is added to the similarity returned from the CCRN. In this way, neighbours are retrieved from the CCRN using a combination of their weight from the CCRN and their sticky boost. By using both negative and positive contributions to control the boost levels of neighbours we are assuring that only neighbours that are *consistently* performing well will be promoted. Of course the sticky boost levels (as well as the short-term profile, or temporary activations in the CCRN), only stay active during a recommendation session.

4 Evaluation

Our preliminary evaluation aims to examine the methods described in this paper, and in particular how (if) the Sticky Layer is helping neighbour selection. The experiments are based on the 100,000 MovieLens Dataset [16], which consists of 100,000 ratings for 1682 movies by 943 users. We select 100 of these users at random as our target users. Each profile consists of a list of movies that the user has seen and a corresponding set of ratings on a scale of 1 - 5, (1 meaning the user did not like the movie and 5 meaning they liked it a lot). This is the same dataset used by Bridge and Kelly [8]. Currently we are using a measure of profile overlap as our profile similarity metric.¹ Since the overlap measure does not require user ratings, we have simplified the MovieLens data by removing all (disliked) profile items with a rating < 3 . Therefore each profile is simply a list of previously liked movies. The average profile size is 88 items. This is also in keeping with our previous research that found that negative preferences are not necessarily as useful as positive ones [5]. Of course, the disliked movies would be retained in a real world system to ensure that they would not be recommended.

4.1 Methodology

In this evaluation we make use of simulated artificial users as a real user trial has not been possible yet. The MovieLens dataset contains genre information (lists

¹ This is in contrast to our previous research and indeed much of collaborative recommendation research where the Pearson Correlation Coefficient is used. However we have found that we actually achieve better results using just overlap. This may mean that the ratings in the MovieLens dataset are not reliable, or the reason may lie elsewhere, but certainly this issue deserves further investigation.

of categories, e.g. “comedy, romance”), for the movies, which we use to simulate user feedback in our evaluation, (though of course genre information is *not* used to generate recommendations). A (simulated) user will select the item with the highest overlap of genre categories with the target item as the best recommended item. The user can select an item so long as it has a genre category overlap > 0 . Otherwise, if all recommended items in a cycle have a genre category overlap of 0, the user will reject them all. Essentially we are modelling a user’s current mood based on genres. Of course there could be many other factors that influence a user’s current mood and preferences, genre information is used here because it was a viable way to do the simulation. A *leave-one-out* test is employed to evaluate the search for specific target items in each evaluation trial. In each trial every item in the user profile is in turn used as the target item (during which time it is removed from the profile). In each cycle 3 recommendations are made, from the profiles of 50 NNs. We evaluate the CCRN and the Sticky Layer methods we propose, using a number of different trials:

CCRN Only ($CCRN$) In this trial we use the CCRN by itself with no Sticky Layer. This performs equally to the standard conversational collaborative recommendation which is to be expected, and serves as our benchmark here.

CCRN with Sticky Layer ($CCRN + S$) Here we examine using the Sticky Layer in conjunction with the CCRN, and consider different combinations of rewards and penalties for neighbour contributions in a recommendation session. We look at four variations, the first where neighbours are only rewarded ($CCRN + S_{rewards}$), the second where the neighbours are both rewarded, and penalised if they contribute to a set of bad items being recommended ($CCRN + S_{rewardsAndPenaliseBadItems}$), the third when the neighbours are both rewarded, and penalised if they don’t contribute to a liked item being recommended ($CCRN + S_{rewardsAndPenaliseGoodItem}$), and finally the fourth where all three are combined ($CCRN + S_{rewardsAndAllPenalties}$). Note that when we refer to $CCRN + S$ we are referring to all four techniques in general. Refer to Section 3.1 for more details.

Bootstrapped Sticky Layer(S) Here we use the Sticky Layer by itself without the CCRN, except at the very start of a session when we use the CCRN until 5 pieces of information have been added to the Sticky Layer, in order to avoid it having to choose neighbours completely at random. So although the Sticky Layer in these experiments receives some help at the start, we would argue that this bootstrapping is minimal. Note that we would not expect this technique to perform better than the ones always using the CCRN since it cannot use any of the main long-term profile information contained in the CCRN. We test it by itself here to gain a more exact idea of how it performs. As with the combined CCRN and Sticky Layer approach we again test four variations of this technique ($S_{rewards}$, $S_{rewardsAndPenaliseBadItems}$, $S_{rewardsAndPenaliseGoodItem}$, and $S_{rewardsAndAllPenalties}$), (and S refers to all four techniques in general).

CCRN with Persistent Sticky Layer ($CCRN + S_{Persist}$) This uses a Sticky Layer that is persistent across sessions. Instead of clearing the sticky

information after a given session it is retained for future sessions. This is testing how strong our theory is that a user’s current preferences are controlled by her current mood and that certain information is only relevant for a limited amount of time.

Random ($SS - CR_{Random}$) Here we run a trial where the NNs are selected at random and the results are averaged over 10 runs (SS-CR = C-CR if neighbours are selected at random). We use this as a final lower benchmark.

In each version the system is evaluated according to three different search criteria. Each one measures the average session length, which equates to recommendation quality.

Item the actual target item needs to be found (recommended) in order to count as a success, i.e. in order for a session to be completed satisfactorily.

Genre an item with the same genre categories as the target item needs to be found.

SimGenre an item with similar genre categories to that of the target needs to be found. (We define two sets of genre categories to be similar if they have an overlap value of ≥ 0.5).

4.2 Results

The results for each of our different system variations, against each of our different success criteria, are presented in Table 1. They report the average length of session needed before the success criterion is found (smaller values are better). The results show that there is little or no difference between the trials that use a Sticky Layer with the CCRN ($CCRN + S$), and the baseline trial that only uses the CCRN, ($CCRN$). This is disappointing as we would have expected that the Sticky Layer would improve the quality of the neighbours selected and ultimately reduce the session lengths. However, this may be due to the Sticky Layer not being given enough influential weight.

When we compare the trial where the Sticky Layer is persistent ($CCRN + S_{Persist}$) to when it is not, we find that it performs significantly worse. This is to be expected and further supports our argument that a user’s *current* preferences can be different from her more long-term static preferences [5].

If we compare how the Boostrapped Sticky Layer (S), performs (with little influence from the CCRN) two things are clear. Firstly, penalising users because they did not contribute to a liked item being recommended is too harsh a strategy to be effective. This is evident from looking at the techniques that penalise neighbours for not contributing to a liked item being recommended ($S_{rewardsAndPenaliseGoodItem}$ and $S_{rewardsAndAllPenalties}$) which perform no better than the random technique. Conversely, if we don’t penalise users in this manner in the Sticky Layer ($S_{rewards}$ and $S_{rewardsAndPenaliseBadItems}$), we can achieve results that far outperform the random technique, and that are similar to $CCRN + S$ techniques. So although the addition of the Sticky Layer is not improving the CCRN results, there is certainly some value in using it. Moreover,

Table 1. Experimental Results

	Item	Genre	SimGenre
<i>CCRN</i>	162.2	53.92	11.98
<i>CCRN + S_{rewards}</i>	161.91	53.95	12.01
<i>CCRN + S_{rewardsAndPenaliseBadItems}</i>	160.62	53.46	11.93
<i>CCRN + S_{rewardsAndPenaliseGoodItem}</i>	163.22	53.89	11.98
<i>CCRN + S_{rewardsAndAllPenalties}</i>	162.05	53.43	11.89
<i>S_{rewards}</i>	162.98	53.69	11.97
<i>S_{rewardsAndPenaliseBadItems}</i>	164.47	54.36	12.7
<i>S_{rewardsAndPenaliseGoodItem}</i>	220.94	65.37	12.21
<i>S_{rewardsAndAllPenalties}</i>	220.28	64.97	12.78
<i>CCRN + S_{Persist}</i>	180.52	60.19	13.03
<i>SS - CR_{Random}</i>	223.85	68.68	14.94

the very fact that the Sticky Layer can achieve results comparable the CCRN may be promising in dealing with the *cold-start* problem faced by collaborative recommender systems [17]. The problem arises when a new user starts to use the system and there is no preference information for her yet, making recommendation difficult. Similarly, our temporary CCRN activations (short-term profile) and Sticky Layer are both empty too at the start of a recommendation session.

5 Conclusions

We propose that *sticking* with users that are contributing positively to the recommendation process is a novel and useful way of increasing recommendation accuracy and reducing session length. Our preliminary evaluations have shown that Sticky Layer recommendation performs better than random recommendation and confirms that our approach has merit. While our results are not overwhelmingly successful, we propose that it may have some value in the cold-start problem of initial recommendations (even if it requires some minimal bootstrapping, it could help speed up the delay while a user is building up enough preference history).

The second contribution of this work is the implementation of a novel data structure called the Collaborative Case Retrieval Net (CCRN) for retrieving NNs for collaborative recommenders. This work follows on from earlier work in the CBR and ACF fields by Hayes *et al.* [9] and Delany *et al.* [13]. CCRNs offer an efficient and fast way of retrieving useful neighbours and allow the incorporation of additional context data into the recommendation process. We demonstrate this with the sandwiching of the Sticky Layer onto the CCRN.

In the future we will investigate the individual usefulness of the Sticky Layer and the temporary and permanent components of the CCRN, by adjusting their relative weights. We will also investigate the use of a decay function to prevent certain users from leading the recommendation process after they have outlived their usefulness. Finally a live user trial is in the pipeline.

References

1. Rafter, R., Bradley, K., Smyth, B.: Automated Collaborative Filtering Applications for Online Recruitment Services. In: Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Trento, Italy, Springer-Verlag (2000) 363 – 368
2. Bridge, D.: Towards Conversational Recommender Systems: A Dialogue Grammar Approach. In: Proceedings of the Workshop in Mixed-Initiative Case-Based Reasoning, Workshop Programme at the Sixth European Conference in Case-Based Reasoning, Aberdeen, Scotland (2002) 9–22
3. Goker, M., Thompson, C.: The Adaptive Place Advisor: A Conversational Recommendation System. In: Proceedings of the 8th German Workshop on Case Based Reasoning, Lammerbuckel, Germany (2000)
4. Aha, D.W., Breslow, L.A., Muñoz-Avila, H.: Conversational Case-based Reasoning. *Applied Intelligence* **14**(1) (2001) 9–32
5. Rafter, R., Smyth, B.: Conversational Collaborative Recommendation - An Experimental Analysis. *Artificial Intelligence Review* **24**(3–4) (2005) 301 – 308
6. McGinty, L., Smyth, B.: Evaluating Preference-Based Feedback in Recommendation Systems. In: Proceedings of the 13th Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2002), Limerick, Ireland, Springer-Verlag (2002) 209–214
7. Hayes, C., Cunningham, P.: Context Boosting Collaborative Recommendations. *Knowledge-Based Systems* **17**(2–4) (2004) 131–138
8. Bridge, D., Kelly, J.P.: Ways of Computing Diverse Collaborative Recommendations. In Wade, V., Ahsman, H., Smyth, B., eds.: *Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer (2006) 41–50
9. Hayes, C., Cunningham, P., Smyth, B.: A Case-Based Reasoning View of Automated Collaborative Filtering. In: ICCBR '01: Proceedings of the 4th International Conference on Case-Based Reasoning, Vancouver, British Columbia, Canada, Springer-Verlag (2001) 234–248
10. Billsus, D., Pazzani, M.: A Hybrid User Model for News Story Classification. In: Proceedings of 7th International Conference On User Modelling (UM99), Banff, Canada (1999) 99–108
11. Widiantoro, D.H., Ioerger, T.R., Yen, J.: An Adaptive Algorithm for Learning Changes in User Interests. In: Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM '99), Kansas City, Missouri, ACM Press (1999) 405–412
12. Lenz, M., Auriol, E., Manago, M. In: *Diagnosis and Decision Support*. Springer (1998) 51–90
13. Delany, S., Cunningham, P., Coyle, L.: An Assessment of Case-Based Reasoning for Spam Filtering. *Artificial Intelligence Review* **24**(3–4) (2005) 359–378
14. Richter, M.M.: Introduction. In Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S., eds.: *Case-Based Reasoning Technology, From Foundations to Applications*, Springer (1998) 1–16
15. Doyle, D., Loughrey, J., Nugent, C., Coyle, L., Cunningham, P.: FIONN: A Framework for Developing CBR Systems. *Expert Update* **8**(1) (2004) 11–14
16. GroupLens: MovieLens Dataset. (<http://www.grouplens.org/>)
17. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating Word of Mouth. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, Denver, Colorado, ACM Press (1995) 210–217