Title	Conversational collaborative recommendation - an experimental analysis
Author(s)	Rafter, Rachael; Smyth, Barry
Publication date	2005-11
Publication information	Artificial Intelligence Review, 24 (3-4): 301-308
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/4639
Publisher's statement	The final publication is available at www.springerlink.com
Publisher's version (DOI)	http://dx.doi.org/10.1007/s10462-005-9004-8

Conversational Collaborative Recommendation - An Experimental Analysis

Rachael Rafter and Barry Smyth
Adaptive Information Cluster, Smart Media Institute,
School of Computer Science and Informatics, University College Dublin, Ireland.
({rachael.rafter, barry.smyth}@ucd.ie*)*

Abstract. Traditionally, collaborative recommender systems have been based on a single-shot model of recommendation where a single set of recommendations is generated based on a user's (past) stored preferences. 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. Such interactions can range from high-level dialogues with the user, possibly in natural language, to more simple interactions where the user is, for example, asked to indicate a preference for one of k suggested items. Importantly, the feedback attained from these interactions can help to differentiate between the user's long-term stored preferences, and her current (short-term) requirements, which may be quite different. We argue that such interactions can also be beneficial to collaborative recommendation and provide experimental evidence to support this claim.

Keywords: collaborative filtering, recommender systems, conversational recommendation

1. Introduction

Until recently collaborative recommender systems have been styled on a *single-shot* model of recommendation, where a single set of recommendations is generated based entirely on a user's stored preference information, for example (Konstan et al., 1997; Rafter et al., 2000; Rafter and Smyth, 2001). The process is a non-interactive one; no current information is sought from users at recommendation time regarding what they are looking for, and the recommendations are based solely on what the users have liked or disliked in the past. In content-based recommender systems however, a shift is emerging towards a more *conversational* model of recommendation where the users are engaged interactively during the recommendation process, for example (Bridge, 2002; Goker and Thompson, 2000; Shimazu, 2002). The level of interaction between the user and system can range from high-level dialogues with the user, possibly in natural language, to low-level interactions where, for example, the user simply needs to select the most appropriate item from a set of k items. This extra information can importantly provide feedback that distinguishes between a user's *short-term* requirements, and more general or *long-term* preferences. A user's short-term requirements may be a specialisation of her long-term preferences, or may even represent current information needs that are quite different from her usual preferences.

Take, for example, a movie recommender where a given user typically likes to watch foreign language movies and documentaries. There will be times when the user particularly wants to watch an Italian movie because, for example, she has recently travelled there, or even times when the user feels like a change and would prefer to watch a comedy. The traditional single-shot recommender will be able to identify that the user generally likes foreign language movies and documentaries. However, it will not be able to recognise the user's current mood which, as in the above examples, may deviate somewhat from her usual movie preferences. A conversational recommender system on the other hand will be more sensitive to such mood changes because it can exploit immediate feedback from the user that reflects her current mood, in order to guide the recommendation process.

Although we are seeing content-based recommender systems shift from the single-shot recommendation model to the conversational model, there has been little published work that examines the role that such feedback can play in collaborative recommender systems. We argue that collaborative recommendation can equally benefit from conversational feedback. In fact, given that collaborative recommendation is a query-less technique, and thus no indication is given by the user at recommendation time as to her current requirements, conversational feedback may be particularly beneficial to collaborative recommendation, if the user's immediate needs are to be satisfied.

Our goal in this paper then, is to investigate the usefulness of conversational feedback in collaborative recommendation. We are also interested in deciding how to incorporate short-term interests gleaned from conversations with the user into collaborative recommendation. In (Rafter and Smyth, 2004) we presented initial results which indicated that conversational collaborative recommendation was indeed performing better than its traditional single-shot counterpart. In this paper we extend these previous findings though a significant range of experimentation. We test the robustness of our conversational collaborative recommendation model across a wide range of varied parameters, evaluation measures and different levels of noise. We will begin by describing the collaborative recommendation process, the conversational model of recommendation and then, how they can be fitted together to produce a more accurate conversational collaborative recommender system. We will then provide details of an extensive experimental study that examines the validity of such a conversational collaborative recommender system within the movie domain.

2. Background

In this section we give a brief description of collaborative recommendation and of the conversational model of recommendation.

2.1. Single-shot Collaborative Recommendation

Single-shot (traditional) collaborative recommendation (SS-CR) is a *content-free* recommendation strategy, that relies on information about users and their preferences for different items. It is based on the premise that similar users like similar things and it exploits correlations between what a given *target user* and other users have liked in the past, to make recommendations of new items to the target user. For example, consider a holiday recommender system. A *content-based* (non-collaborative) version might contain cases of different holidays described by features such as location, price, and duration. It will then identify similarities between different cases to find a new holiday for a user often based on what she has liked in the past, or that best matches her query. A collaborative recommender on the other hand, would contain profiles that record (unique) identifiers for holidays (no content information is needed) that the user has taken in the past, and, usually a set of ratings that describe how much the user liked (or disliked) each one. The traditional single-shot collaborative recommendation process is depicted in Figure 1.

It is important to note here, that the user profile only contains information about the user's long-term preferences. There is no information pertaining to what the user may be *specifically* interested in when a recommendation is being made. Some researchers therefore have proposed user profiles where both the *short-term* preferences, as well as the long-term preferences of the user are represented, for example (Billsus and Pazzani, 1999; Balabanovic, 1998; Goker and Thompson, 2000; Koychev and Schwab, 2000; Widyantoro et al., 1999).

2.2. Conversational Model of Recommendation

The conversational recommendation model has recently become a popular approach in content-based recommender system research. Conversational recommender systems can be classified according to the type of feedback that they elicit from the end-user. The most obvious model involves natural language dialogues between the system and the user, for

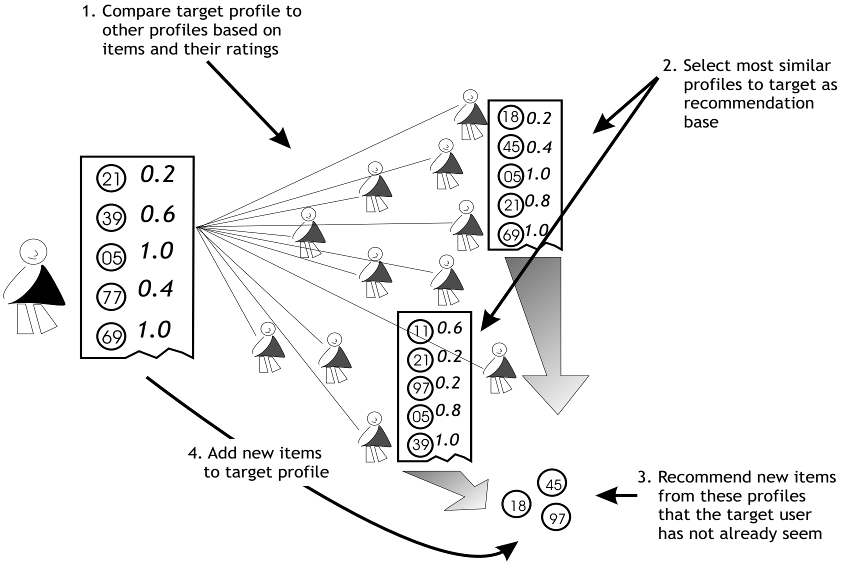
1. Compare target profile to other profiles based on items and their ratings
2. Select most similar profiles to target as recommendation base
21 0.2
39 0.6
05 1.0
77 0.4
69 1.0
18 0.2
45 0.4
05 1.0
21 0.8
69 1.0
11 0.6
21 0.2
97 0.2
05 0.8
39 1.0
4. Add new items to target profile
45
18
97
3. Recommend new items from these profiles that the target user has not already seem

Figure 1. Traditional Single-shot Collaborative Recommendation

example (Goker and Thompson, 2000). However, many conversational systems use feedback instead as a simpler way of extracting necessary information from the user, and we concentrate on this type of system here.

Feedback strategies are often classified according to whether they operate at *feature-level* or *case-level*, or, according to the cost to the user, *low-cost* or *high-cost*. Strategies that operate at the feature level include *value elicitation* and *tweaking*. Such strategies ask that the user provide feedback about the features of a recommended item, for example, indicating in a holiday recommender that she would prefer a holiday that was cheaper (price), or a holiday that was longer (duration). Strategies operating at the case-level include *rating-based feedback* and *preference-based feedback*. In contrast to feature-level feedback, here the user is asked to provide feedback about cases, or items, as wholes. For example, in a holiday recommender, a user might be presented with k initial holiday recommendations and asked to order them according to her preference, or select one that best matches what she wants, without considering their individual features. With respect to the cost to the user, feature-level strategies tend to be high-cost, while case-level strategies tend to be low-cost. For a full description of all these strategies, see (McGinty and Smyth, 2002).

3. Towards Conversational Collaborative Recommendation

We have so far described collaborative recommendation and various styles of conversational recommendation. As we already mentioned, our intention is to investigate how best to employ user feedback from conversational interactions to improve collaborative recommendation.

3.1. Basic Algorithm

Our research focuses on the preference-based feedback style of conversational recommendation, described in Section 2.2. The process is similar to that of the traditional single-shot algorithm, in that nearest neighbours to a target user are selected based on similarities between their profiles, and recommendations are generated from the neighbour profiles. However there are some key differences. An important feature of conversational collaborative recommendation (C-CR) is that feedback from the user at recommendation time can help distinguish between her *short-term* and *long-term* preferences. As mentioned in Section 2.1, the ability to accurately classify user preferences into short-term and long-term preferences has been shown to be important in recommendation.

In C-CR, cycles of k recommendations are made to the user (*cycle phase*) and she is asked to indicate which 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 added to the profile as part of the user's short-term preferences. This process is repeated, with new items being recommended to the user each time based on the updated profile, until the *target item* is found, (i.e. the item that best matches her current needs) or there are no more possible recommendations. With each addition to the short-term preferences, the selection of nearest neighbours is more finely tuned towards the user's current requirements. The process is detailed in Figure 2.

3.2. Update Variations

We now detail some of the ways in which the short-term information can be used to update the profile, representing the target user's short-term preferences. Two strategies are considered, and later in Section 4, evaluated as alternative update strategies for C-CR.

3.2.1. *ST$^{+/-}$ (Positive and Negative Update)*

The first update strategy, ST$^{+/-}$ (short-term positive and negative), groups the short-term information into positives and negatives. If the target user selects one of the k recommended items in the current cycle

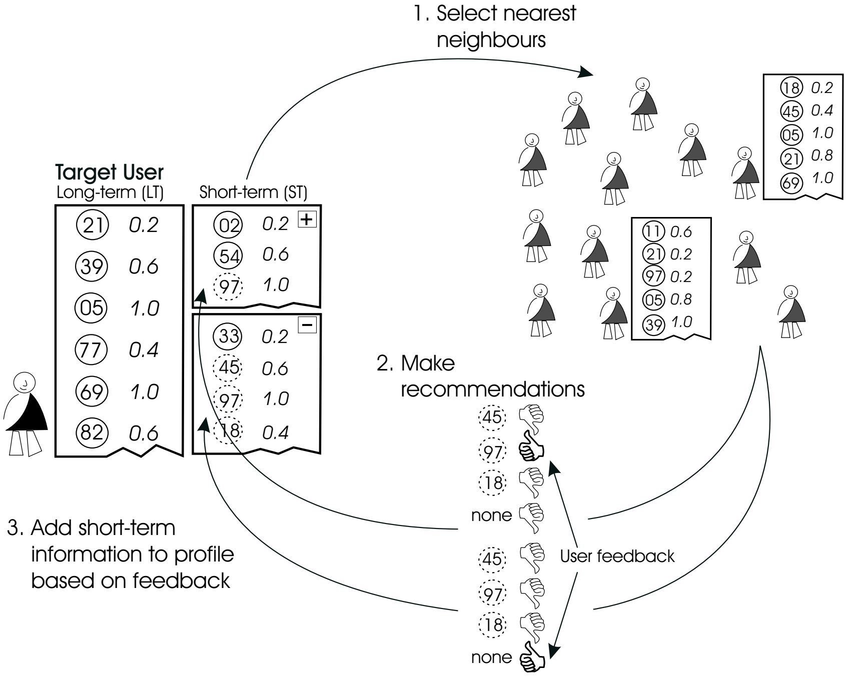

Figure 2. Conversational Collaborative Recommendation

as preferable, it is added to the short-term part of the profile for positive items (ST$^+$). If the user decides that none of the recommendations are suitable, all *k* recommendations are added to the short-term part of the profile for negative items (ST$^-$). In either case, the items that are added to the short-term part of the profile are excluded from being recommended in future cycles. Note that if the user selects one of the items, the other remaining *k-1* items are not added as negative items, as it is possible that the user liked them also.

With each addition to the short-term part of the profile, the selection of nearest neighbours is more finely tuned to the user's current needs. The selection is directed towards users that have liked the items in the target user's ST$^+$ preferences, and, towards users that have disliked the items in the target user's ST$^-$ preferences. In other words, the algorithm looks for neighbours that share both liked and disliked items with the target user's short-term preferences. The correlation between the target user's long-term (LT) preferences and the neighbour profiles is still taken into account as in the single-shot algorithm. Equation 1 shows how the weight for a neighbour *n*, (*w(n,t)*) to a target user *t* is calculated. Note that t^{LT} refers to the long-term part of the target user's profile, and t$^{ST^+}$ and t$^{ST^-}$ refer to the positive and the negative short-term parts of the profile respectively. Finally, *hMean* refers to the harmonic mean.

$$w(n,t) = correl(n, t^{LT}) * hMean(overlap(n, t^{ST^+}), overlap(n, t^{ST^-})) \tag{1}$$

3.2.2. *ST^+ (Positive Update Only)*
Our second update strategy, ST^+ (short-term positive) is similar to the $\mathrm{ST}^{+/-}$ strategy, but considers only positive short-term preferences. Equation 2 shows the calculation of the weight. The short-term part of the profile is only updated if the user selects one of the k recommended items during the cycle phase as preferred. The items that the user disregards during the cycle-phase, are not taken into account, and the algorithm looks only for neighbours that share liked items with the target user's short-term preferences. However, all items that have been recommended to a user (whether added to the short-term part of the profile or not) are excluded from being recommended again, as in the $\mathrm{ST}^{+/-}$ strategy. The reasoning behind this update strategy is that, although it follows that when a target user and a neighbour both like the same items, the target user should like other items that the neighbour has liked, it does not necessarily follow the other way round; that a target user and a neighbour both dislike the same items does not always mean that they will like the same items.

$$w(n,t) = correl(n, t^{LT}) * overlap(n, t^{ST^+}) \tag{2}$$

4. Experimental Evaluation

We have so far described how conversational, preference-based feedback can be used in collaborative recommendation. In order to evaluate our assertion that such feedback can be exploited to better guide collaborative recommendations towards what the user is currently seeking, we have carried out a preliminary evaluation. In our evaluation we are interested in the comparative performance (efficiency in finding suitable recommendations) of the different collaborative recommendation algorithms discussed in Sections 2.1 and 3. In particular we want to look at how the conversational algorithms perform when compared to the single-shot algorithm, and which of our two conversational update strategies (ST^+ and $\mathrm{ST}^{+/-}$) work best. More specifically, we are examining how many recommendations need to be made before we find a satisfactory one. (In Section 4.3 we will expand more fully on what we mean by a satisfactory recommendation).

4.1. Setup

Using the MovieLens [1] dataset we have taken the 2100 largest user profiles, randomly selecting 100 of them as 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). The average profile size is 355 items. The MovieLens dataset also contains *genre* information for the movies (lists of categories, for example “action” or “comedy, romance”), which we use for our evaluation (though of course genre information is not actually used in collaborative recommendation which is content-free).

4.2. Methodology

In this evaluation we make use of a simulated artificial user as a real user trial was not feasible. A *leave-one-out* test is employed to evaluate the search for specific *target items* in each evaluation trial. Three evaluation trials are performed for each target user; in each trial every item in the user profile that is rated highly (rating >= 3, i.e. items the user has liked) is in turn used as the target item (during which time it is temporarily removed from the profile):

SS-CR: traditional single-shot collaborative recommendation, (Section 2.1). A list of all possible recommendations is made for the target user, and the position in the list where (if) the first satisfactory recommendation occurs.

C-CR$^{+/-}$: conversational collaborative recommendation (Section 3.1) using both positive and negative short-term information, described in Section 3.2.1.

C-CR^{+}: conversational collaborative recommendation using only positive short-term information, described in Section 3.2.2.

For the two C-CR algorithms, three recommendations are made to the target user in each cycle, until a satisfactory recommendation is found (if one is found). As in the SS-CR algorithm, the total number of recommendations made up to this point is noted. We simulate user preference feedback by assuming that in each cycle, the user will select a recommended movie if it has the same, or to a lesser extent, similar (overlapping) genre categories as the target item. For efficiency purposes, a cycle limit of 100 cycles (300 recommendations) is imposed.

If the target item is not found within 300 recommendations by both the C-CR algorithms and the single-shot algorithm, the results for that item are discarded.

4.3. Varying Success Criteria

In total, over the 100 target users we collected results in each evaluation trial from 27943 target items (except those discarded due to the cycle limit). The results (Sections 4.3.1 and 4.3.2) are presented in two ways. The first set of results examines the average number of recommendations needed to be made before a satisfactory recommendation is found, by the three algorithms: SS-CR, C-CR$^+$ and C-CR$^{+/-}$. The second set of results looks at the number of cases when either of the C-CR algorithms beat the SS-CR algorithm, and *vice versa*.

In order to examine the comparative performance of the different algorithms, we need a measure of what constitutes a *satisfactory* recommendation. We term this measure the *success criterion*, in other words, the criterion that needs to be satisfied by any possible recommendation to count as a success. In the following sections, we present the results using a number of different success criteria.

4.3.1. *Item As Success Criterion*

The first success criterion we consider is the $success_{item}$ criterion, where the actual target item needs to be found (recommended) in order to count as a success. The results are shown in Figures 3 (a) and (b). Figure 3 (a) shows the average number of recommendations needed to be made before a success is encountered for the SS-CR, C-CR$^+$ and C-CR$^{+/-}$ algorithms. Figure 3 (b) shows the percentage of times that the C-CR$^{+/-}$ and C-CR$^+$ algorithms outperform (and tie with) the traditional SS-CR algorithm. In total 54% of the target items were included in the $success_{item}$ results after the cycle limit was imposed.

It is clear from these results that eliciting feedback from users at recommendation time significantly improves recommendation performance. If we look at the performance of the C-CR$^+$ algorithm (Figure 3 (a)), it achieves a reduction in the number of recommendations needed to reach the target item of over 20%, over the single-shot (SS-CR) algorithm. When we look at actual 'wins' (Figure 3 (b)), this in fact translates to the C-CR$^+$ algorithm outperforming the SS-CR technique in 57% of the cases, (wins). In 12% of the cases the algorithms tied, which means that in 69% of the cases C-CR$^+$ performs at least as well, or better, than the SS-CR technique. Similar results are found using the C-CR$^{+/-}$ algorithm.

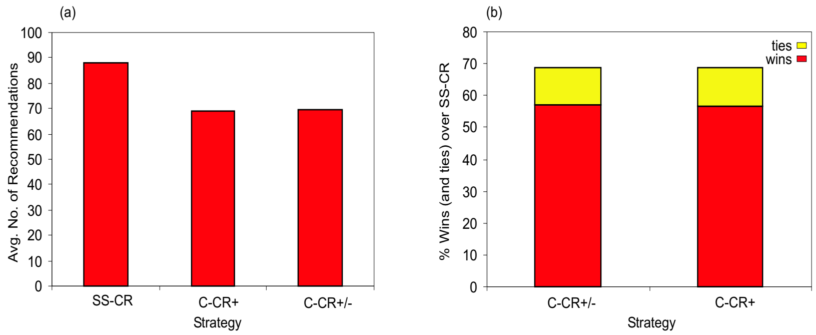

Figure 3. (a) Average No. of Recommendations Required with $success_{item}$ Measure, and (b) % Wins (and ties) over SS-CR with $success_{item}$ Criterion

Of course, the $success_{item}$ criterion is particularly strict, the artificial user only modelling users that are extremely particular about that for which they are looking. As a result, the number of recommendations needed to be made before a suitable recommendation is found, with any of the algorithms, is considerably long. Clearly, we do not propose that real users would tolerate lists of such length, nor, that many real users are likely to be so particular. However strict the $success_{item}$ criterion is though, it is still a useful way to measure comparative performance as it is a concrete success criterion.

4.3.2. *Relaxing the Success Criterion*

We have so far described the $success_{item}$ criterion in which the exact target item needs to be recommended in order to count as a success. This simulates the situation where the user is very particular about the movie they want to watch. However, since other users will be less particular, or more open about the movie they want to watch, we also evaluate performance in each trial using two supplementary success criteria, $success_{genre}$ and $success_{simGenre}$. The $success_{genre}$ criterion, requires that we recommend an item with exactly the same genre categories as the target item, simulating the situation where a user is less particular about the movie they want to watch but they have clear idea about the *kind* of movie they want to watch. Clearly, such a success criterion need not exclusively be concerned with genres. Other features such as the director could also be considered here.

The results are shown in Figures 4 (a) and (b), again detailing the average number of recommendations needed before a successful one is found, and the cases where the C-CR algorithms outperform (and tie with) the SS-CR algorithm, when using the $success_{genre}$ criterion,

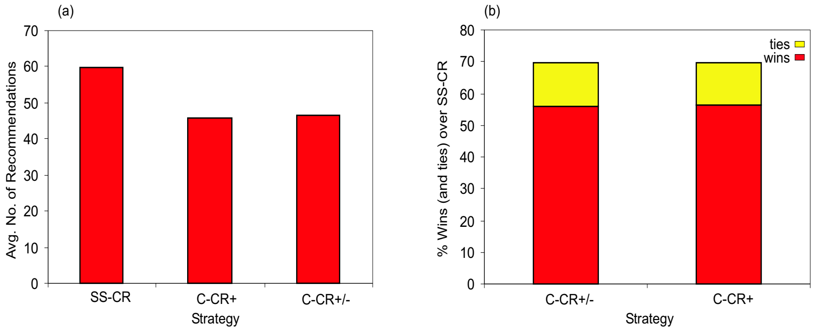

Figure 4. (a) Average No. of Recommendations Required with $success_{genre}$ Criterion, and (b) % Wins (and ties) over SS-CR with $success_{genre}$ Criterion

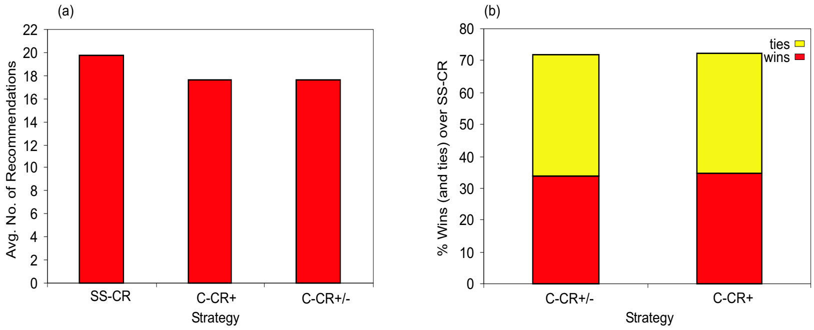

Figure 5. (a) Average No. of Recommendations Required with $success_{simGenre}$ Criterion, and (b) % Wins (and ties) over SS-CR with $success_{simGenre}$ Criterion

respectively. In total 88% of the target items were included in the $success_{genre}$ results after the cycle limit was imposed.

Our last success criterion, the $success_{simGenre}$ criterion, requires that we recommend an item that has *similar* genre categories (i.e. that its genre categories overlap with the target item's genre categories by at least a given threshold; in these experiments the threshold was set to 0.5, however we have also experimented with thresholds of 0.3, 0.7 and 0.9 which perform predictably and linearly, see Figure 6). This is basically a weaker version of the $success_{genre}$ criterion. In total 99% of the target items were included in the $success_{simGenre}$ results after the cycle limit was imposed. The results are shown in Figures 5 (a) and (b).

When we look at the $success_{genre}$ and $success_{simGenre}$ results (Figures 4 and 5) when the success criteria are relaxed, we see a similar pattern in performance improvement to that seen when we use the $success_{item}$ criterion, although the gain from incorporating short-

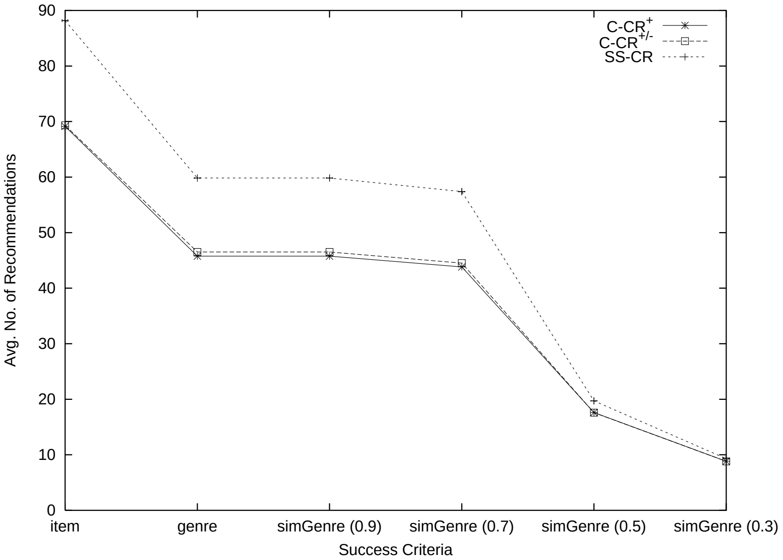

Figure 6. Summary of Performance using SS-CR, C-CR^{+} and C-CR$^{+/-}$ with each of $success_{item}$, $success_{genre}$ and $success_{simGenre}$ Criteria

term information into the profile is less pronounced. Indeed using the $success_{simGenre}$ criterion results in a large number of ties between the C-CR algorithms and the SS-CR algorithm. This is because it is easier for all the algorithms to find a successful recommendation and hence there is less room for the performance to differ. However importantly the number of recommendations needed to be made is significantly reduced as the success criteria are less strict. We believe, that probably the most realistic success criteria is $success_{genre}$ where the user is not so rigid in her acceptance of recommendations as suitable, but still has a clear idea of the boundaries that define where a suitable recommendation might be found.

4.3.3. *Success Criteria Summary*

The results presented so far show that both of the C-CR algorithms improve recommendation performance in the sense that they both reduce the session length with the user (number of recommendations made before a success) significantly. We summarise the results across all the different success criteria in Figure 6, which also includes the supplementary results obtained from using different thresholds for genre similarity (thresholds of 0.3, 0.7 and 0.9 in addition to the original threshold of 0.5) with the $success_{simGenre}$ criterion.

The results also show that when the ST^{-} preferences are filtered into the neighbour selection process in the C-CR$^{+/-}$ algorithm, the results are not any better than when only ST^{+} preferences are used. We mentioned in Section 3.2.2 that although users who share liked items are likely to make good recommendation partners for each other, it does not necessarily follow that users who share disliked items will make good recommendation partners for each other. This is one possibility as to why we see no significant improvement in the results that include the negative short-term preference information. Clearly though, this matter deserves further attention. Since there is no significant difference between the $\mathrm{C\text{-}CR}^{+}$ and $\mathrm{C\text{-}CR}^{+/-}$ results, we only consider the $\mathrm{C\text{-}CR}^{+}$ strategy in the remaining part of our evaluation.

4.4. Noise Results

Until now, we have assumed that users will always make reliable choices when presented with possible recommendations during the cycle phase in conversational collaborative recommendation. However, as users do not generally know exactly what they are searching for, this is not necessarily the case, and often users will choose a recommendation as preferable over the others even though this is in fact leading the search away from the item with which they will finally be satisfied. We would like our algorithm to be robust enough to deal with these misleading user choices.

Therefore, in order to simulate misleading user choices, we add noise at different levels for the C-CR algorithms. At each different level, n, the simulated artificial user selects a *random* recommendation as preferable in the cycle phase every n cycles. Thus for example at $n = 2$, the user is selecting a random recommendation in every second cycle. We do not add noise for the SS-CR algorithm as we want to see how well our C-CR algorithms perform against the SS-CR algorithm, even when they are affected by noise. The results are shown in Figures 7, 8 and 9. In each figure, the average number of recommendations needed before a successful one is found with the $\mathrm{C\text{-}CR}^{+}$ algorithm is plotted at different noise level inputs $(= 2, 5, 10, 15)$. Note that we have also included in each of these graphs the results with a noise level of 1, i.e. where a random choice is made each time. This serves to show that it is not simply because there is extra information in the short-term part of the profile that the conversational algorithms perform better. It is only when we add appropriate, or, good information to the short-term part of the profile that the conversational algorithm performs well. The results are graphed in comparison to those for the (stable) SS-CR algorithm. Figure 7 shows the results using the $success_{item}$ success criterion, while Figures 8 and 9 show the results using the $success_{genre}$ and the $success_{simGenre}$ algorithms, respectively. Note here that although

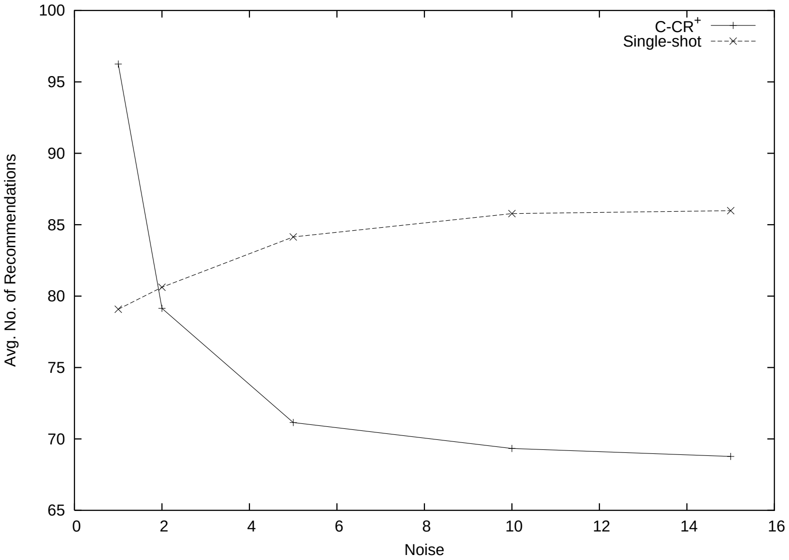

Figure 7. Performance of C-CR$^+$ with different levels of noise against SS-CR with $success_{item}$ Criterion

the SS-CR algorithm was not affected by noise, the values still vary slightly. This is due to the cycle limit imposed as mentioned earlier, where results are excluded if both the C-CR and SS-CR algorithm do not find the target item within 300 recommendations, and means that the SS-CR values are averaged over a different number of cases at each noise level. Importantly though, the C-CR and SS-CR algorithms are compared to each other over the same set of items at each different noise level.

The results show that the C-CR$^+$ algorithm is robust enough to withstand considerable levels of noise input. In fact, using the $success_{item}$ criterion, the C-CR$^+$ algorithm outperforms the SS-CR algorithm across the entire range of noise levels. It even outperforms the SS-CR algorithm when random feedback is given every second cycle (noise level = 2). Using the $success_{genre}$ criterion, the C-CR$^+$ algorithm stands up to noise at levels of 5, 10 and 15, but not 2. However, the situation where a user will choose a random (or incorrect) recommendation every second cycle is not one that we see as very likely, and so we believe this is still a very promising result. Even with the $success_{simGenre}$ criterion which has seen the smallest margin of performance gain by the C-CR algorithms over the SS-CR algorithm in our experimentation, the C-CR$^+$ algorithm still holds up well.

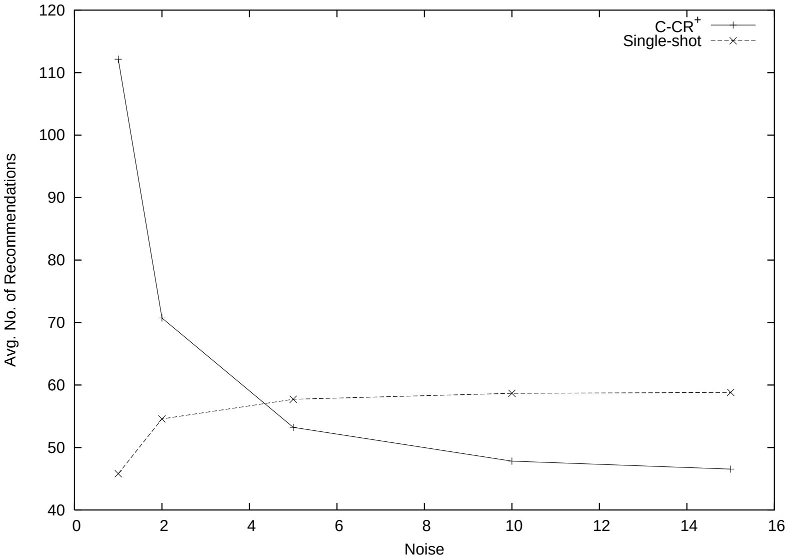

Figure 8. Performance of C-CR$^+$ with different levels of noise against SS-CR with $success_{genre}$Criterion

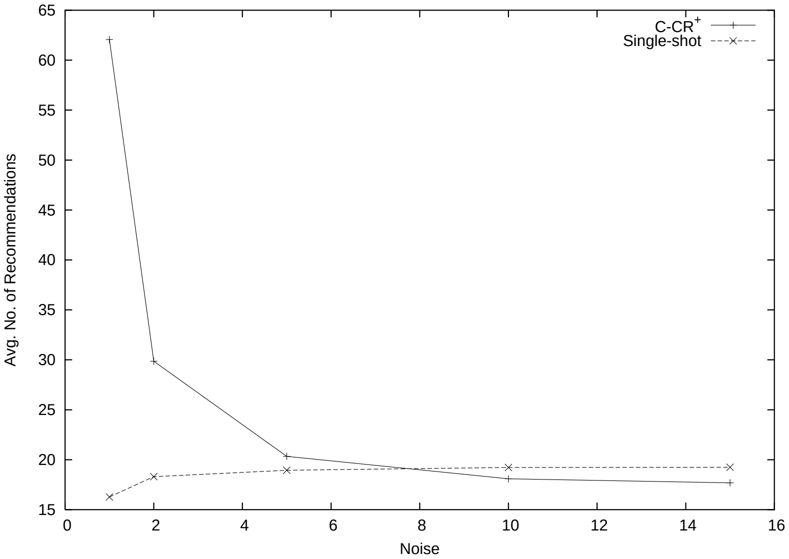

Figure 9. Performance of C-CR$^+$ with different levels of noise against SS-CR with $success_{simGenre}$ Criterion

5. Conclusions and Future Work

We have proposed that feature-based feedback gleaned from conversational style recommendation can be used to improve the performance of collaborative recommendation, and we have shown preliminary experimental evidence to support this. In the future we plan to investigate further the reasons why the use of negative short-term information does not show any performance improvements. We also intend to look at item-based collaborative filtering (Sarwar et al., 2001) which identifies similarities between items rather than between users. This could then be used to identify items in the user's LT profile that are similar to those in the user's ST preferences, thus giving us a larger base of *important* items to work with. It could also be used to select recommendations in a given cycle that are similar to items the user has preferred in previous cycles.

Acknowledgements

This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361.

Notes

[1] GroupLens Research Group http://www.grouplens.org

References

Balabanovic, M.: 1998, 'An Interface for Learning Multi-topic User Profiles from Implicit Feedback'. In: *Proceedings of AAAI Workshop on Recommender Systems.* Madison, Wisconsin, U.S.A.

Billsus, D. and M. Pazzani: 1999, 'A Hybrid User Model for News Story Classification'. In: *Proceedings of 7th International Conference On User Modelling (UM99).* Banff, Canada, pp. 99–108.

Bridge, D.: 2002, '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, pp. 9–22.

Goker, M. and C. Thompson: 2000, 'The Adaptive Place Advisor: A Conversational Recommendation System'. In: *Proceedings of the 8th German Workshop on Case Based Reasoning.* Lammerbuckel, Germany.

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

Koychev, I. and I. Schwab: 2000, 'Adaptation to Drifting User's Interests'. In: *Proceedings of ECML2000 Workshop: Machine Learning in New Information Age*. Barcelona, Spain, pp. 39–45.

McGinty, L. and B. Smyth: 2002, 'Comparason-Based Recommendation'. In: *Proceedings of the Sixth European Conference on Case-Based Reasoning (ECCBR-02)*. Aberdeen, Scotland, pp. 575–589, Springer-Verlag.

Rafter, R., K. Bradley, and B. Smyth: 2000, 'Automated Collaborative Filtering Applications for Online Recruitment Services'. In: *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*. Trento, Italy, pp. 363 – 368, Springer-Verlag.

Rafter, R. and B. Smyth: 2001, 'Passive Profiling from Server Logs in an Online Recruitment Environment'. In: *Proceedings of IJCAI Workshop on Intelligent Techniques for Web Personalization (ITWP2001)*. Seattle, Washington, U.S.A., pp. 35–41.

Rafter, R. and B. Smyth: 2004, 'Towards Conversational Collaborative Recommendation'. In: *Proceedings of the 15th Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2004)*. Castlebar, Mayo, Ireland.

Sarwar, B. M., G. Karypis, J. A. Konstan, and J. Riedl: 2001, 'Item-based collaborative filtering recommendation algorithms'. In: *Proceedings of the 10th International World Wide Web Conference (WWW10)*. Hong Kong, pp. 285–295, ACM Press.

Shimazu, H.: 2002, 'ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops'. *Artificial Intelligence Review* **18**(3-4), 223–244.

Widyantoro, D. H., T. R. Ioerger, and J. Yen: 1999, '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, U.S.A., pp. 405–412, ACM Press.