<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Towards Conversational Collaborative Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authors(s)</strong></td>
<td>Rafter, Rachael; Smyth, Barry</td>
</tr>
<tr>
<td><strong>Publication date</strong></td>
<td>2004-09</td>
</tr>
<tr>
<td><strong>Conference details</strong></td>
<td>The 15th Irish Conference on Artificial Intelligence and Cognitive Science (AICS 04), Castlebar, Mayo, Ireland, 8-10 September, 2004</td>
</tr>
<tr>
<td><strong>Item record/more information</strong></td>
<td><a href="http://hdl.handle.net/10197/4640">http://hdl.handle.net/10197/4640</a></td>
</tr>
</tbody>
</table>
Towards Conversational Collaborative Filtering *

Rachael Rafter and Barry Smyth

Adaptive Information Cluster, Smart Media Institute,
Department of Computer Science, 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 are 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 deep dialogues with the user that may involve natural language dialogues, 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 preliminary experimental evidence in support of this.

1 Introduction

Until recently collaborative recommender systems have been styled on a single-shot model of recommendation, where a single set of recommendations are generated based entirely on a user’s stored preference information, for example [1, 2]. 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 [3]. The level of interaction can range from high-level natural language dialogues, 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 their long-term preferences, or may even represent current information needs that are quite different from their usual preferences.

* This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361
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, they have recently travelled there, or even times when the user feels like a change and would prefer 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 their 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 (CR) 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 interested in deciding how to incorporate short-term interests gleaned from conversations with the user into collaborative recommendation. 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 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 conversational models of recommendation.

2.1 Single-shot Collaborative Recommendation

Single-shot (traditional) collaborative recommendation (SS-CR) 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 other 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, duration, etc. 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.

![Diagram of traditional single-shot collaborative recommendation](image)

**Fig. 1.** Traditional Single-shot Collaborative Recommendation

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 et. al. [4] separated profiles into short-term goals (likely to change quite often) where the user is committed to finding a particular piece of information at the current time, and long-term goals (stable) which refer to the long-term goals of the user. Balabanovic [5] discusses the need to separate "I like this item" from "Show me more like this item" in recommender systems. He points out that, indicating that a document is relevant to a topic or user profile, and indicating that more similar documents should be recommended in the next iteration, clearly do not always coincide. A user can enjoy an article about a particular item, without desiring all following interactions to be solely concerned with that item. Others, for example [6], have pointed out that in news articles, for example, a user’s preferences may drift over time.
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 example [7]. 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 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 [8].

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 technique, 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 they are 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). 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.

![Fig. 2. Conversational Collaborative Recommendation](image)

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 strategies to the traditional single-shot algorithm.

**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 during the cycle-phase 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 is suitable, all $k$ recommendations are added to the short-term part of
the profile for negative items \((ST^-)\). 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 \(n\) 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.

\[
w(n, t) = \text{correl}(n, t^{LT}) \times \text{harmonicMean}(\text{overlap}(n, t^{ST^+}), \text{overlap}(n, t^{ST^-}))
\] (1)

**ST\(^+\) (Positive Update Only)** Our second update strategy, \(ST^+\) (short-term positive) is similar to the \(ST^+/^-\) 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 selected 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 for neighbours that have only shared liked items with the target user’s short-term preferences. The reasoning behind this 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.

\[
w(n, t) = \text{correl}(n, t^{LT}) \times \text{overlap}(n, t^{ST^+})
\] (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 exactly what the user is currently seeking, we have carried out a preliminary evaluation.
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 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 (lists of categories) for the movies, which we use for our evaluation (though of course genre information is not actually used in collaborative recommendation as it 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 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, described in Section 2.1. A list of all possible recommendations is made for the target user, and the position in the list where (if) the target item occurs is noted.
- **C-CR\(^{+/−}\)**: conversational collaborative recommendation (Section 3.1) using both positive and negative short-term information, described in Section 3.2.
- **C-CR\(^+\)**: conversational collaborative filtering using only positive short-term information, described in Section 3.2.

In the two C-CR algorithms, we simulate user preference feedback by assuming that in each cycle, the user will select a recommended movie if it has the same or similar genre categories as the target item. Three recommendations are made to the target user in each cycle, until the target item is found. Again, the total number of recommendations made before finding the target item is noted. For efficiency purposes, a cycle limit of 100 cycles is employed. If the target item is not found within 100 cycles the results for that item are discarded.

4.3 Results

In total we collected results in each evaluation trial from 27943 target items (except those discarded due to the cycle limit). Figures 3 and 4 show the results.

Figure 3 shows the average number of recommendations that need to be made to find the target item for the SS-CR, C-CR\(^{+/−}\) and C-CR\(^+\) techniques. Figure 4 shows the % of times that the C-CR\(^{+/−}\) and C-CR\(^+\) techniques outperform (and tie with) the traditional SS-CR algorithm.

It is clear from these results that eliciting feedback from users at recommendation time significantly improves recommendation performance. the C-CR\(^+\)

\(^1\) GroupLens Research Group http://www.grouplens.org
technique achieves a reduction in the number of recommendations needed to reach the target item of 50%, over the single-shot technique. When we look at actual wins, this in fact translates to the C-CR+ technique outperforming the SS-CR technique in 52% of the cases, (wins). In 36% of the cases the techniques tied which means that in 88% of the cases C-CR+ performs at least as well, or better, than the SS-CR technique.

In fact, it is also true that when either of the C-CR techniques lose to the SS-CR technique it is by a small amount, however, when they win, they win by a lot. This is shown in Figures 5 and 6 which graph the comparative average number of recommendations made when a particular technique wins. The gain is especially true with the C-CR+ technique. We can see in Figure 5, that when the C-CR+ technique beats the standard SS-CR technique the average number of recommendations needed is reduced from approximately 270 recommendations...
to 110 recommendations. However, when the SS-CR technique beats the C-CR $^+$ technique, the reduction is not so substantial, from approximately 120 recommendations down to 85 recommendations.

![Fig. 5. Comparison of Gain From Wins between SS-CR and C-CR$^+$](image)

The results also show that when the ST$^-$ preferences are filtered into the neighbour selection process in the C-CR$^{+/\sim}$ technique, the results are not quite as good as when only ST$^+$ preferences were used, although they are clearly better than the SS-CR technique. As mentioned in Section 3.2 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.
It should also be noted here that this evaluation is a rather strict one, where only one particular recommendation (the target item) will count as a success. We believe however, that it is often the case that the user will be satisfied with a range of items rather than just one particular item.

5 Conclusions and Future Work

We have proposed that feature-based feedback gleaned from conversational style collaborative filtering 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 look at the performance of the C-CR algorithms when we relax the success criteria - when there is a range of recommendation that count as successes, rather than just a single recommendation. We also intend to look at item-based collaborative filtering [9] 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 for the cycle-phase that are similar to items the user has preferred in previous cycles.

References