Artif Intell Rev (2006) 25:79–95
DOI 10.1007/s10462-007-9023-8

Enhancing the diversity of conversational collaborative recommendations: a comparison

John Paul Kelly · Derek Bridge

Published online: 18 August 2007

Abstract In conversational collaborative recommender systems, user feedback influences the recommendations. We report mechanisms for enhancing the diversity of the recommendations made by collaborative recommenders. We focus on techniques for increasing diversity that rely on collaborative data only. In our experiments, we compare different mechanisms and show that, if recommendations are diverse, users find target items in many fewer recommendation cycles.

Keywords Recommender systems · Collaborative recommendations · Diversity

1 Introduction

Recommender systems suggest products, services or information sources to their users (Resnick and Varian 1997; Riedl and Konstan 2002). They differ in the way they find the items they recommend (Balabanović and Shoham 1997):

Content-based systems: The system stores a description of each available item. A user describes the item that she wants as a query or she describes the kinds of items that she likes as entries in a user profile. The system compares the user's descriptions against the store of item descriptions and recommends items that match.

Collaborative systems: Item descriptions are not used. A user's profile stores user opinions against item identifiers. The system compares other users with the active user and recommends items that were liked by users whose profiles are similar to the active user's profile.

Recommender systems differ also by the extent to which they engage in dialogue with the user (Bridge et al. 2006):

J. P. Kelly (✉) · D. Bridge
Department of Computer Science, University College Cork, Western Road, Cork, Ireland
e-mail: jpk2@cs.ucc.ie

D. Bridge
e-mail: d.bridge@cs.ucc.ie

Table 1 A ratings matrix

	Ann	Bob	Col	Deb	Edd	Flo
Cape Fear	⊥	⊥	3	5	5	5
Naked Gun	3	2	⊥	2	4	⊥
Aliens	⊥	5	⊥	⊥	2	4
Star Wars	4	2	3	3	3	⊥
Taxi Driver	⊥	⊥	3	4	3	⊥

Single-shot systems: In response to a user request (and, where appropriate, submission of a user query), the system delivers a set of recommendations to the user. Each request is treated independently of previous ones.
Conversational systems: Users elaborate their requirements over the course of an extended dialogue. On receiving a set of recommendations, the user might refine her query; or she might supply feedback on the recommended items. Her refined query or feedback influences the next set of recommendations.

Conversational systems can more easily adapt their recommendations to the user's short-term interests. By dint of mood changes or other special circumstances, short-term interests may not coincide with long-term interests. Your normal preference for cop flicks, for example, may be superceded today by a desire for an Italian travelogue in preparation for up-coming travel.

There is a mature body of research on conversational *content-based* systems, dealing with issues such as question selection and ordering (Doyle and Cunningham 2000; Schmitt 2002) and processing user feedback (Burke et al. 1997; McGinty and Smyth 2002; Reilly et al. 2004). But research into *collaborative* systems has focused on single-shot recommenders. The work of Rafter and Smyth (2004) is a recent exception. In Sect. 3 we describe conversational collaborative recommenders based on the one in (Rafter and Smyth, 2004). In Sect. 4, we discuss how to enhance the diversity of the recommendations produced by conversational collaborative recommenders. In Sect. 5, we compare diversity-enhanced conversational collaborative recommenders with a single-shot recommender and with a conversational collaborative recommender that takes no special steps to enhance the diversity of its recommendations. Then, in Sects. 6 and 7, we define and compare several different mechanisms for enhancing the diversity of collaborative recommendations. But first, in Sect. 2, we summarise the operation of the class of collaborative recommenders used in this work.

2 Collaborative recommenders

In a collaborative recommender, given m items, $I = \{i\colon 1, \ldots, m\}$, and n users, $U = \{u\colon 1, \ldots, n\}$, preferences are represented using a $m \times n$ matrix of ratings $r_{i,u}$. Users may explicitly supply their item ratings or the system may obtain an implicit rating from observing user actions. Note that it is possible and common that $r_{i,u} = \perp$, signalling that the user has not yet rated that item. An example of a ratings matrix for movies is shown as Table 1. Each column in the matrix is a user's long-term profile. We will write u^{LT} for the item identifiers that have non-$\perp$ ratings in user u's long-term profile. For example, $\text{Bob}^{LT} = \{\text{Naked Gun, Aliens, Star Wars}\}$.

There are many ways of building collaborative recommenders, most of which are compatible with the research reported in this paper. Here we describe just the one we have implemented; for details, see (Herlocker 2000):

– The similarity $w_{u_a,u}$ between the active user u_a and each other user, $u \neq u_a$, is computed using Pearson correlation, $correl(u_a, u)$, over their co-rated items, $C =_{\text{def}} u_a^{LT} \cap u^{LT}$. In the case of users who have co-rated fewer than 50 items (who, even if their ratings are very similar, are likely to be poor predictors), Pearson correlation is devalued by a significance weight (Herlocker 2000), $sig(u_a, u)$.

$$w_{u_a,u} =_{\text{def}} correl(u_a, u) \times sig(u_a, u) \tag{1}$$

$$correl(u_a, u) =_{\text{def}} \frac{\sum_{i \in C}(r_{i,u_a} - \bar{r}_{u_a})(r_{i,u} - \bar{r}_u)}{\sigma_{u_a}\sigma_u} \tag{2}$$

$$sig(u_a, u) =_{\text{def}} \begin{cases} \frac{|C|}{50} & \text{if } |C| < 50 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

$\bar{r}$ denotes a mean value and σ denotes a standard deviation, and these are computed on co-rated items only. For later parts of this paper, it is important to appreciate that $correl(u_a, u)$ and hence $w_{u_a,u}$ can be positive, zero or negative.

– After computing the similarity between u_a and each other user, u, the N (in our case, 20) *nearest neighbours* are selected, i.e. the N for whom $w_{u_a,u}$ is highest.

– For each item i that has not been rated by u_a but has been rated by at least one of the neighbours, u_a's rating for i is predicted, p_{i,u_a},

$$p_{i,u_a} =_{\text{def}} \bar{r}_{u_a} + \frac{\sum_{u=1}^{N}(r_{i,u} - \bar{r}_u)w_{u_a,u}}{\sum_{u=1}^{N} w_{u_a,u}} \tag{4}$$

This is a weighted average of the deviations between the neighbours' ratings for item i and their mean ratings. If p_{i,u_a} goes out of the range of legal ratings, it is rounded to the nearest end-point of the range (J. Herlocker, personal communication, 2002).

– These items are then sorted into descending order of p_{i,u_a}. This is the order in which items will be recommended. For example, if in a single-shot system we want to recommend three items, then the first three items in this sorted list are selected.

3 The CCR$^+$ and CCR$^\pm$ systems

In 2004, Rafter and Smyth described their conversational collaborative recommender system: the system recommends items to the user; the user gives feedback on the recommendations; and the feedback influences the next set of recommendations (Rafter and Smyth 2004). We use CCR$^+$ and CCR$^\pm$ to designate our implementation of their idea (Bridge and Kelly 2005).

In CCR$^+$ and CCR$^\pm$, the active user has a long-term profile (based on a column in the ratings matrix), u_a^{LT}, as do all other users. But, for the duration of her interaction with the system, the active user also has two short-term profiles, $u_a^{ST^+}$ and $u_a^{ST^-}$.

Initially, the short-term profiles are empty and the first set of k (typically, three) recommendations is made in the fashion described in Sect. 2. At this point, the system solicits user feedback. If she so wishes, the user can terminate the dialogue, with or without having

chosen an item for purchase or consumption. On the other hand, if she wishes to continue the dialogue, she can optionally provide feedback of one of the following two forms:

- She can indicate which recommended item best matches her short-term interests. If she does, the selected item's identifier is added to her short-term positive profile, $u_a^{ST^+}$. Nothing is done with the other items.
- She can indicate that none of the recommended items adequately meets her short-term interests. If she does, the identifiers of all the recommended items are added to her short-term negative profile, $u_a^{ST^-}$.

If the dialogue has not been terminated, the system then recommends another set of items. New recommendations never repeat ones made previously in this dialogue. But, additionally, through the way it computes user similarity, the system attempts to steer new recommendations towards the kind of items in $u_a^{ST^+}$ and away from the kind of items in $u_a^{ST^-}$; see below for details. This recommendation-and-feedback cycle continues until either the user finds an item she wishes to consume, she abandons the dialogue having found no such item, or the system can make no fresh recommendations.

This form of feedback is referred to as case-level, preference-based feedback (McGinty and Smyth 2002). It operates at the level of items (cases) and not, for example, at the level of item attributes (such as movie genre); and the user expresses preferences but gives no information about what it is that she likes or dislikes about the items. It is important to distinguish the *feedback* a user may provide during a dialogue from the *rating* she might supply for an item she ultimately consumes. The feedback is indicative of short-term interests, used to influence recommendations made within the course of the current dialogue; the ratings are taken to represent long-term interests.

Of course, in a real system, a user may not be able to provide feedback based solely on an item identifier, such as a movie title. If there are descriptive attributes (e.g. film genre and director's name), reviews, clips and trailers, these may need to be shown to the user too.

It remains to say how $u_a^{ST^+}$ and $u_a^{ST^-}$ influence subsequent recommendations. The idea in conversational collaborative recommending is that the selection of nearest neighbours is "…directed towards users that have liked the items in the target user's [short-term positive profile], and towards users that have disliked items in the target user's [short-term negative profile]" (Rafter and Smyth 2004, p. 152). We partition u^{LT} into two: the likes and the dislikes. The likes (the long-term positive profile) we denote by u^{LT^+} and this set is compared with $u_a^{ST^+}$. The dislikes (the long-term negative profile) we denote by u^{LT^-} and this set is compared with $u_a^{ST^-}$. For example, in the MovieLens dataset, whose rating scale is 1–5, u^{LT^+} contains items rated 3 or above; u^{LT^-} contains items rated below 3. When finding neighbours, the similarities between users will no longer be based just on the significance-weighted Pearson correlation between their long-term profiles. Now, users whose long-term positive profiles contain items in the active user's short-term positive profile will receive a boost, as will users whose long-term negative profiles contain items in the active user's short-term negative profile.

We will describe Rafter and Smyth's boosting formulae, and we will describe ways of fixing two potential problems with these formulae.

- In Rafter and Smyth's version of CCR$^+$, the more a user's long-term profile overlaps with the active user's short-term positive profile, the greater the boost:

$$w_{u_a,u} =_{\text{def}} correl(u_a, u) \times sig(u_a, u) \times overlap(u_a^{ST^+}, u^{LT^+}) \tag{5}$$

– In their version of CCR$^{\pm}$, overlaps with the active user's short-term positive and negative profiles are combined by H, the Harmonic mean:

$$w_{u_a,u} =_{\text{def}} correl(u_a, u) \times sig(u_a, u) \times H(overlap(u_a^{ST^+}, u^{LT^+}), overlap(u_a^{ST^-}, u^{LT^-})) \quad (6)$$

where $H(x, y) = \frac{2xy}{x+y}$.

Given that $u_a^{ST^+}$, $u_a^{ST^-}$, u^{LT^+} and u^{LT^-} are simply sets of item identifiers, the *overlap* function is defined as the size of the intersection of its two arguments (R. Rafter, personal communication 2004).

However, a problem is that empty intersections are quite likely to happen.

– In CCR^{+}, if the user repeatedly rejects all recommended items, then $u_a^{ST^+}$ will remain empty until there is a cycle in which the user sees an item that suits her short-term interests, at which point that item is added to $u_a^{ST^+}$.
– In CCR$^{\pm}$, if the user repeatedly rejects all recommended items, then $u_a^{ST^-}$ grows but $u_a^{ST^+}$ remains empty; if, instead, the user repeatedly selects an item from those recommended, then $u_a^{ST^+}$ grows but $u_a^{ST^-}$ remains empty. The intersection will become non-empty only when there has been at least one cycle in which an item is added to $u_a^{ST^+}$ and at least one cycle in which items have been added to $u_a^{ST^-}$.
– In both systems, even when the short-term profiles are non-empty, they are quite small so there is still a high risk that the overlap with another user's long-term profile will be empty. (The two users may have co-rated some items, so $correl(u_a, u)$ is not zero, but they may not have co-rated the particular items that are in the active user's short-term profile.)

Without a fix, the upshot is that, during the early stages of the dialogue, many or all users will have zero overlap and hence zero similarity to the active user: the neighbours will be determined only by the system's tie-breaking mechanism. Hence, Rafter & Smyth do not use the overlap when it is zero (R. Rafter, personal communication 2005) and, with the same effect, in our implementations of CCR^{+} and CCR$^{\pm}$ we use the following:

$$overlap(A, B) =_{\text{def}} \max(1, |A \cap B|) \quad (7)$$

which defaults to 1 when the intersection is empty. (Other definitions are possible without making any major difference to the results, e.g. $1 + |A \cap B|$.)

A second problem is that $correl(u_a, u)$ can be negative. On occasion, there may be so few positively correlated neighbours that negatively correlated users are among those with the highest values for $correl(u_a, u)$. When this is the case, Eqs. 5 and 6 would multiply this negative number by the overlap, which is positive. Far from boosting the similarity of a user with high overlap, the resulting value for $w_{u_a,u}$ will be a negative number of greater magnitude and so the user will be less likely to be a neighbour. Rafter and Smyth overcome this problem by excluding negatively-correlated users from the set of neighbours. However, excluding negatively-correlated users will (slightly) narrow the set of items that may be recommended. We felt that this was undesirable given that the goal of the system is to make recommendations that do not necessarily reflect the user's normal long-term interests. We overcome the problem of negatively-correlated users by *adding* overlap values to $correl(u_a, u)$, rather than multiplying. (We tried some better-motivated schemes, including some that used normalisation of overlap values, as well as Rafter and Smyth's approach, but in experiments they were outperformed by our own approach.)

In summary then, we define $w_{u_a,u}$ as follows:

– In CCR$^+$:

$$w_{u_a,u} =_{\text{def}} correl(u_a, u) \times sig(u_a, u) + overlap(u_a^{ST^+}, u^{LT^+}) \tag{8}$$

– In CCR$^\pm$:

$$\begin{aligned} w_{u_a,u} =_{\text{def}} & \; correl(u_a, u) \times sig(u_a, u) \\ & + overlap(u_a^{ST^+}, u^{LT^+}) + overlap(u_a^{ST^-}, u^{LT^-}) \end{aligned} \tag{9}$$

Although the version of CCR$^+$ and CCR$^\pm$ that we have reported here outperformed a number of variants, the two systems perform poorly compared with a single-shot system that makes repeated recommendations with no user feedback (see Sect. 5). A more radical innovation is needed.

4 The CCR$^+$-Div and CCR$^\pm$-Div systems

This section introduces the CCR$^+$-Div(b, k) and CCR$^\pm$-Div(b, k) systems. In their names, Div indicates a concern for the diversity of recommendations; b and k are parameters, which are explained below.

For *content-based* recommender systems, the argument has been convincingly made that items should be selected for *diversity* (relative to each other) as well as *similarity* (to the query or the user's profile) (Smyth and McClave 2001). Too much similarity between the recommended items (e.g. three Woody Allen movies) can be undesirable. But, when recommendations are diverse, if the user is not satisfied with the most highly recommended item, for example, the chances of her being satisfied with one of the alternative recommendations is increased.

There is a body of research that addresses diversity for *content-based* recommenders, e.g. (Linden et al. 1997; Bradley and Smyth 2001; Smyth and McClave 2001; Bridge and Ferguson 2002; McSherry 2002; Shimazu 2002; Pu et al. 2006). While its importance is recognised (Herlocker et al. 2004), it is only now that we are seeing the first work that attempts to improve the diversity of the items recommended by *collaborative* recommenders. Specifically, apart from our own work, we are aware only of Ziegler et al.'s work on book recommendations (2005). Neglect of diversity may be because collaborative recommenders can provide *serendipitous* recommendations (Herlocker 2000). Serendipitous recommendations are pleasing recommendations for unexpected items; on occasion, they may increase diversity. However, we hypothesise that a more direct concern for diversity may be important, especially in *conversational* collaborative systems: diverse recommendations increase the chances of positive feedback (where an item is preferred over the others), and this helps the system target the recommendations towards the user's short-term interests.

To investigate this, we implemented the Bounded Greedy selection algorithm (henceforth BG) from (Smyth and McClave 2001). To recommend k items, BG finds bk items, where b is a positive integer greater than one. In (Smyth and McClave 2001), these are the bk items that are most similar to the query (using content-based recommending). Here, they are the bk items with the highest prediction values p_{i,u_a} (where neighbours are computed by the CCR$^+$ or CCR$^\pm$ systems). From these bk items, BG selects k to recommend to the user. It selects

Candidates ← *bk* items recommended by CCR^{+} (or $\text{CCR}^{\pm}$)
$R \leftarrow \{\}$
for $j \leftarrow 1$ to k **do**
  best ← the $i \in$ *Candidates* for which $Quality(i, R)$ is highest
  insert *best* into R
  remove *best* from *Candidates*
end for
return R

Fig. 1 The Bounded Greedy selection algorithm

the k in a greedy fashion, based on ones selected so far; see Fig. 1 (adapted from (Smyth and McClave 2001)).

In our variant of the algorithm, the quality of item i relative to the result set so far R is defined as follows:

$$Quality(i, R) =_{\text{def}} (1 - \alpha) \times p_{i,u_a} + \alpha \times RelDiv(i, R) \tag{10}$$

i.e. it is a weighted combination of the predicted rating for item i and the diversity we will attain if we include i in R. α is a factor that allows the importance of the predicted rating and relative diversity to be changed. In this paper, we only investigate the case where the two factors are given equal weight. Hence, we normalise both factors so that they fall in [0, 1] and we then use $\alpha = 0.5$.

Diversity relative to the result set so far is defined as the average distance between i and the items already inserted into R:

$$RelDiv(i, R) =_{\text{def}} \begin{cases} 1 & \text{if } R = \{\} \\ \frac{\sum_{j \in R} dist(i,j)}{|R|} & \text{otherwise} \end{cases} \tag{11}$$

This will lie in [0, 1] provided each $dist(i, j)$ lies in [0, 1].

This leaves the issue of how to measure distance *between items* in Eq. 11. In (Smyth and McClave 2001), the distance between items is the inverse of the *content-based* similarity. If item descriptions are available, the same approach can be used to enhance the diversity of collaborative recommendations. For example, Ziegler et al. (2005) use taxonomic knowledge in their system. But we choose to proceed on the assumption that item descriptions are not available. We enhance diversity using a measure of distance that is calculated using *collaborative data only*, i.e. we use only the ratings matrix.

Our approach to distance using collaborative data only is based on the following heuristic:

Two items are different if the users who rated them are different.

The intuition is that the community of users who have rated item i have a certain set of tastes. The more the membership of the community who rated item i differs from the membership of the community who rated item j, the more likely i and j satisfy different tastes and are different kinds of items. For example, according to this heuristic, a movie that is liked exclusively by adolescent males is likely to be distant from one that it liked exclusively by middle-aged women. (We stress, however, that, just as we are not using content-based data, we are not using demographic data either: our ways of computing distance will make use only of the ratings matrix.)

There are numerous ways to make this informal heuristic into something concrete that can be implemented. In Sects. 6 and 7 of this paper, we describe and compare four such

	Ann	Col	Deb
Cape Fear	0	1	1
Naked Gun	1	0	1
	↘	↓	↙
Hamming distance:		2	
Normalised:		$\frac{2}{3}$	

	Ann	Col	Deb
Cape Fear	0	1	1
Taxi Driver	0	1	1
	↘	↓	↙
Hamming distance:		0	
Normalised:		$\frac{0}{3}$	

Fig. 2 Hamming distances

ways. For now, however, we describe only one of these four. The one we describe turns out to be one of the best of the four. It is the method of computing distance that we adopt for our experiments in Sect. 5, where we compare diversity-enhanced recommendations with standard recommendations.

In the method we adopt, the distance between two items is inversely related to the size of the intersection of the sets of neighbours who rated the two items. This definition of distance can be computed quite efficiently using bit-vectors.

In detail, then, we compute $dist(i, j)$ as follows:

- CCR$^+$(or CCR$^\pm$) will already have found u_a's N nearest neighbours.
- For both i and j, we create bit vectors I and J of length N. Digit d in vector I is set if neighbour d has a non-$\perp$ rating for item i; similarly for bits in J.
- $dist(i, j)$ is computed as the Hamming distance between I and J, i.e. a count of the number of positions in which their bits differ. This value is normalised, so that it falls within [0, 1], by dividing it by N, the number of nearest neighbours found.

Figure 2 illustrates this process; it shows Naked Gun to be more different from Cape Fear than Taxi Driver is. In the figure, we take N, the number of nearest neighbours, to be three, and we assume these are Ann, Col and Deb. We take their ratings from Table 1 and set bits to show who rated what.

5 Empirical evaluation I

To evaluate the systems that we have described, we adopt Rafter and Smyth's methodology (Rafter and Smyth 2004), but our datasets differ. They select the 2100 largest user profiles from the '1 Million MovieLens Dataset'; the average profile size for the 2100 users is 355 ratings. We use the entire '100K MovieLens Dataset', which contains profiles for 943 users; the average profile size is 106 ratings, which we think is more realistic.

About 100 user profiles are selected at random from the dataset. Each of these acts in turn as an (artificial) active user. Given the active user's long-term profile, each item in turn is treated as the target item. Each of the systems that we evaluate repeatedly recommends sets of three items to the user until either the target item is one of the recommended items, there have been 100 recommendation cycles, or no further recommendations can be made to this user, whichever comes soonest. If the target item is recommended within 100 cycles, the number of items recommended is recorded. Results are subjected to three-fold cross-validation, with a different 100 active users in each fold.

In each recommendation cycle, the (artificial) user's feedback needs to be simulated. For each movie, the MovieLens datasets record a set of genres, which allows a simple-minded content-based approach. If the target item's set of genres is G_t and a recommended item's set of genres is G_r, we compute $\frac{|G_t \cap G_r|}{|G_t \cup G_r|}$. If all recommended items score zero, then none is taken to match the user's short-term interests, so all the items are inserted into $u_a^{ST^-}$; otherwise, the highest-scoring item (with random tie-breaking) is taken to match the user's short term-interests, so this item is inserted into $u_a^{ST^+}$ and nothing is done with the others.

In the diversity-enhanced systems, we have taken $k = 3$ and $b = 15$. In other words, a set of three items is chosen greedily from the 45 candidates with highest predicted ratings. In previous work, we have partially explored what happens when different values of b are chosen (Bridge and Kelly 2005). There we found better results for $b = 15$ than $b = 5$ and $b = 10$. Of course, it does not follow that results will continue to improve with ever larger values of b: at some point, the set of candidates will be so large that it will include items whose predicted ratings are so low that they will not be valuable recommendations. In future work, we need to find the values of b for which this is the case.

In the results reported here, we include a system that uses the BG algorithm but which chooses the k products at random from the bk candidates with the highest predicted ratings. By comparing this random approach with our more informed approach, we can see whether the more informed approach makes any systematic difference beyond that which can be made randomly.

Results are given in Figs. 3–6 and Table 2. We have excluded from these results those systems that use negative short-term profiles, $u_a^{ST^-}$, i.e. the CCR$^{\pm}$ and CCR$^{\pm}$-Div(b, k) systems. Both in the results we have previously published (Bridge and Kelly 2005) and the results in Rafter and Smyth's original paper (Rafter and Smyth 2004), we never encountered a situation where a system that operated on both the short-term positive and negative profiles outperformed the corresponding system that operated on the short term positive profile only. In other words, CCR$^+$ always equals or betters CCR$^{\pm}$; and CCR$^+$-Div(b, k) always equals or betters CCR$^{\pm}$-Div(b, k).

Figure 3 shows, as a percentage of a total of 34759 dialogues, how often the target item was found. In addition to CCR$^+$, and CCR$^+$-Div(15, 3), we show the results for SS-CR, a single-shot recommender, which computes a ranked list of items in the way described in Sect. 2, and recommends them in decreasing order of predicted rating, k (= 3) items at a time. We regard SS-CR as successful if the target item is among all the possible recom-

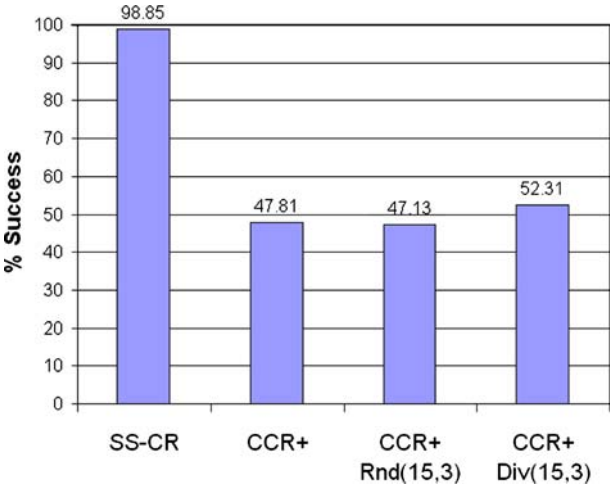

Fig. 3 Success rates

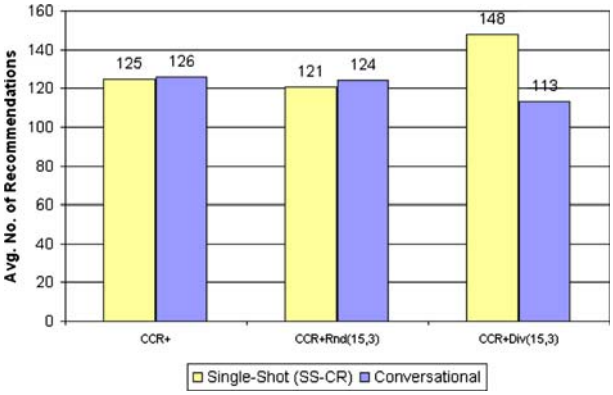

Fig. 4 Average number of recommendations needed to reach the target item

Table 2 Winning and losing margins

	Win	Lose
CCR$^+$	46	24
CCR$^+$-Rnd(15, 3)	43	46
CCR$^+$-Div(15, 3)	48	28

mendations it can make to the active user. The other systems are successful if the target item is recommended within 100 cycles of three recommendations each. Unsurprisingly, SS-CR has by far the highest success rate; encouragingly, the diversity-enhanced system, CCR$^+$-Div(15, 3), has higher success rates than the others.

Figure 4 shows how many items are recommended, on average, before the system recommends the target item. In this figure, each system is compared with SS-CR in cases where both were successful in finding the target item. We see that SS-CR can rival CCR$^+$, which suggests that the user feedback has little value in CCR$^+$. This result is different from the

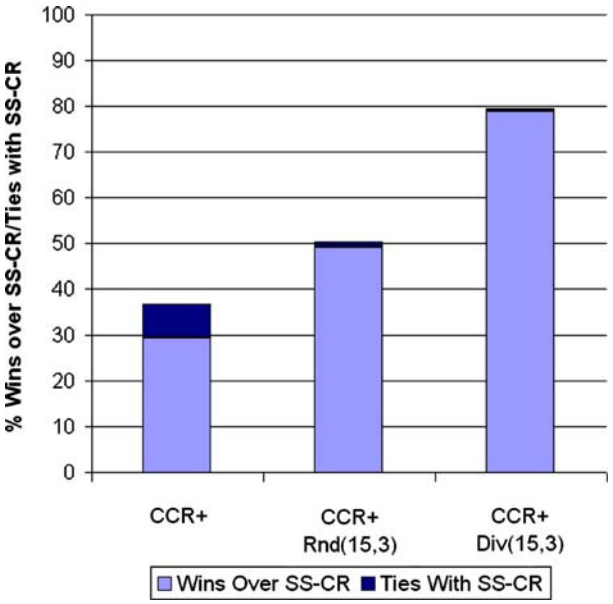

Fig. 5 Wins over SS-CR and ties with SS-CR

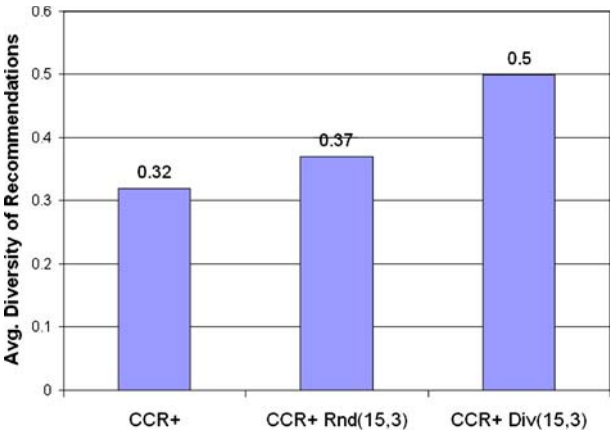

Fig. 6 Average diversity of recommendations

one in (Rafter and Smyth 2004), where CCR$^+$ outperformed SS-CR. The discrepancy can perhaps be explained by the different experimental conditions. In (Rafter and Smyth 2004), the user profiles are the 2100 largest profiles in the '1 Million MovieLens Dataset'; since the profiles contain more ratings on average than they do in our experiments, the probability of useful feedback is increased. In any case, the diversity-enhanced system outperforms SS-CR, which confirms that diverse recommendations can elicit more useful feedback, even when profiles are quite small.

However, all the systems recommend, on average, over 100 items (over 30 recommendation cycles of three items each) before they reach the target. This would clearly not be acceptable in practice. In defence, we note that the experimental methodology is severe for at

least three reasons. First, real users might be satisfied with any one of a set of items, whereas in the experiments there is a single target item each time. Second, the target item may be one to which the user has given a 'negative' rating (i.e. she has given a rating that is below the mid-point of the rating scale); while this helps to model the idea of user short-terms tastes that are not in line with long-term tastes, recommending such items in a dialogue is very challenging. They are likely to have low predicted ratings and, therefore, even allowing for enhanced diversity, are unlikely to be recommended early in the dialogue. Third, the simulated user feedback is so crude that it can sidetrack the conversational recommenders, making them on occasion uncompetitive even with the single-shot system.

Figure 5 and Table 2 compare each system with SS-CR (when both are successful). We see (Fig. 5) that the diversity-enhanced system makes fewer recommendations than SS-CR nearly 80% of the time; the other systems are competitive with SS-CR less than 40% of the time. Then in Table 2 we show winning and losing margins. The table shows, for example, that, when CCR$^+$ wins against SS-CR, it makes on average 46 fewer recommendations and, when CCR$^+$ loses against SS-CR, it makes on average 24 more recommendations. By this measure, the diversity-enhanced system wins by most when it wins, and on average it does not lose by much more than CCR$^+$ when it loses.

Finally, in Fig. 6, we compute for each system the average diversity (i.e. average all-pairs distance) of each set of items it recommends, averaged over all such sets. This confirms that CCR$^+$-Div(b, k) has the best values. On the one hand, this is to be expected: the graph evaluates this system with exactly the measure that it seeks algorithmically to maximise! On the other hand, it shows that the CCR$^+$-Div(b, k) system makes a systematic improvement over a system that selects randomly. All the values may seem low but this is a facet of the averaging; some of the individual recommendation sets may be quite diverse.

We conclude that enhancing the diversity of the recommendations significantly improves the performance of conversational collaborative recommenders. However, in these experiments we adopted one particular definition of item distance. In the next two sections, we propose and compare some alternative definitions.

6 Definitions of item distance

Recall from Sect. 4 that we enhance the diversity of recommendations by greedily selecting k items from bk candidates according to their quality. Quality is a weighted combination of predicted rating and relative diversity (Eq. 10); diversity relative to the result set so far is defined as the average distance between a candidate and the items already inserted into the result set (Eq. 11). We took a heuristic approach to defining the distance between a pair of items using collaborative data only: the more the community of users who rated i differs from the community who rated j, the greater is $dist(i, j)$.

In Sect. 5, we took one concrete implementation of this heuristic and evaluated it against standard collaborative recommenders. Specifically, we took the distance between two items to be inversely related to the size of the intersection of the sets of neighbours who rated the two items. Here we will refer to that definition as 'Hamming Distance over NN', where NN stands for nearest neighbours. As we noted in Sect. 4, this is only one possible definition. Here we compare Hamming Distance over NN to three other definitions, giving four definitions in total.

The four definitions differ on two dimensions:

NN or All Users: The first dimension is the way we choose the set of users on which the communities are defined. Our existing definition, Hamming Distance over NN, confines

attention to the active user's nearest neighbours. We could instead define the communities over the set of *all* users known to the system. In other words, in the former, distance is computed between vectors of length N, where N is the number of nearest neighbours; in the latter, distance is computed over vectors of length n, where n is the size of U. An advantage of the latter is that item-item distances can be computed in advance and cached, needing recalculation only when a new rating arrives. By contrast, if we restrict attention to the nearest neighbours, distances can only be computed on demand, at recommendation time; this is because the set of neighbours depends not just on the active user's long-term profile but also on his or her short-term profiles.

Hamming Distance or Inverse Pearson: The second dimension is the way in which we compute by how much two communities differ. Our existing definition takes no regard of the magnitudes of the ratings. It considers only whether a user has rated an item or not; it counts how many of the users have rated one of the two items but not the other (Hamming distance). We could instead compute the Pearson correlation between the users' ratings of the two items. Effectively, this means computing Pearson correlation between *rows* in the ratings matrix, Table 1. Some collaborative recommenders work on this basis, e.g. (Sarwar et al. 2001). However, we would use its inverse as our measure of item distance, and normalise so that it falls within [0, 1]. This would have the seeming advantage of being sensitive to the magnitudes of the ratings. Note, however, to select k items from bk using the BG algorithm requires $\sum_{i=1}^{k-1}(bk - i)i$ distances to be computed in making each set of k recommendations. It is important for speed of response that these distances be computed quickly. Computing inverse Pearson correlations is likely to be less efficient than the bit-vector implementation of Hamming distance, especially in those systems where the distances cannot be precomputed and cached (see previous point).

Two binary dimensions gives four possible definitions, as shown in Table 3. We emphasise that in all other respects these four systems are identical: only their computation of item–item distance within the definition of relative diversity varies; and we emphasise that the upper left quadrant in Table 3, Hamming Distance over NN, is the definition adopted in Sect. 4 and evaluated in Sect. 5.

7 Empirical evaluation II

We used exactly the methodology of Sect. 5 to compare the four systems that, in their computation of relative diversity, use the different definitions of item distance.

Figure 7 shows, as a percentage of 34759 dialogues, how often the target item was found. As our results showed in Sect. 5, SS-CR is by far the most successful system. However, we see that, of the diversity-enhanced systems, the ones that use Hamming Distance are more successful than those that use Inverse Pearson correlation. In fact, Hamming Distance over NN is 3.22% more successful than Inverse Pearson over NN; and Hamming Distance over All Users is 6.22% more successful than Inverse Pearson over All Users. Computing Hamming Distance over All Users is more successful than computing Hamming Distance over NN, but the gain is marginal: just 0.36%.

Figure 8 shows how many items are recommended, on average, before the system recommends the target item. As in Fig. 4, each system is compared with SS-CR in cases where both were successful in finding the target item. Hamming Distance systems make fewer recommendations on average than Inverse Pearson systems and fewer on average than the single-shot system.

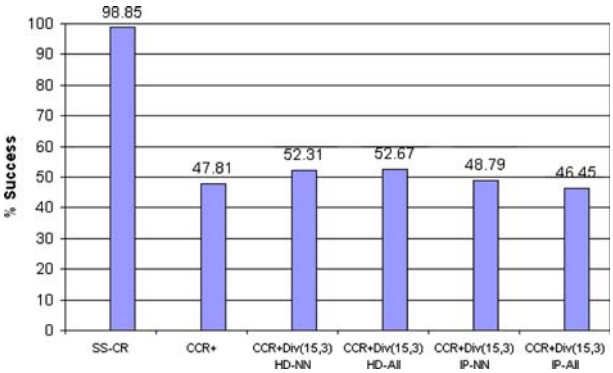

Fig. 7 Success rates

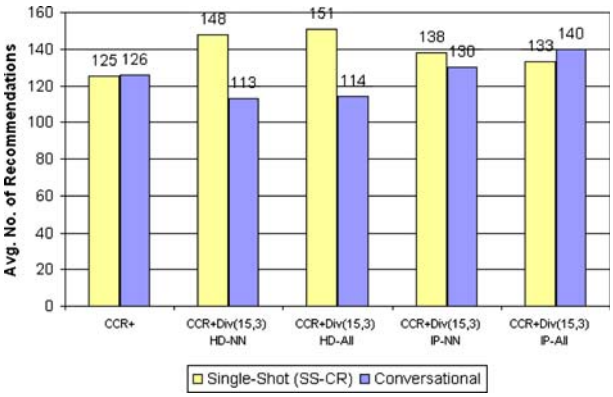

Fig. 8 Average number of recommendations needed to reach the target item

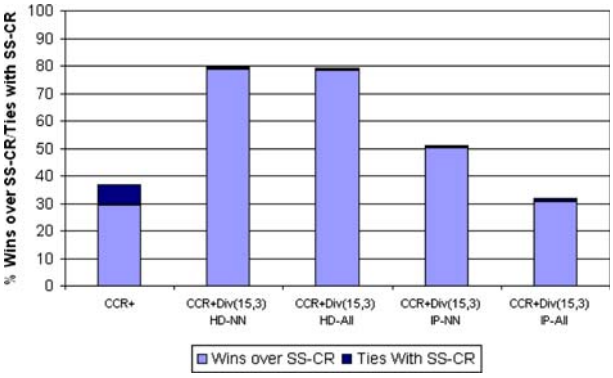

Fig. 9 Wins over SS-CR and ties with SS-CR

Figure 9 and Table 4 compare each system with SS-CR (when both systems have succeeded in finding the target item). Again, the systems that use Hamming Distance are far more successful in beating SS-CR than those that use Inverse Pearson. Table 4 shows that Inverse Pearson over All Users wins over SS-CR by the largest margin, but we know from

Table 3 Four definitions of item distance, using collaborative data only

	Nearest Neighbours	All Users
Hamming Distance	HD-NN	HD-All
Inverse Pearson	IP-NN	IP-All

Table 4 Winning and losing margins

	Win	Lose
Hamming Distance over NN	51	29
Hamming Distance over All Users	56	33
Inverse Pearson over NN	48	33
Inverse Pearson over All Users	59	37

Fig. 9 that it beats SS-CR only 31% of the time. Table 4 shows that it also loses by the largest margin. Hamming Distance over All Users outperforms SS-CR by the largest margin.

Overall, Hamming Distance outperforms Inverse Pearson. This is surprising: ignoring the magnitudes of the ratings is better than taking them into account! We suspect that this is because Hamming Distance, being more discrete, sharpens the definitions of the communities, which is important in our heuristic definition of item diversity, whereas Inverse Pearson, being more continuous, de-emphasises community differences.

In terms of success rates and average numbers of recommendations, there is little to choose between Hamming Distance over NN and Hamming Distance over All Users. What may matter most when choosing between them is efficiency. As we have noted, Hamming Distance over All Users can be precomputed, whereas Hamming Distance over NN cannot. On the other hand, Hamming Distances over All Users, if precomputed, need to be recomputed when new ratings arrive and any one of these distances will take longer to compute than a distance computed over just the NN. More research, focusing on their relative efficiencies, is needed to choose between these two.

8 Conclusions

Building on the seminal work reported in (Rafter and Smyth 2004), we have developed a number of conversational collaborative recommender systems. In all these systems, the selection of neighbours is guided by overlap with the active user's short-term positive and negative profiles. In CCR^{+}-Div(b, k) and CCR$^{\pm}$-Div(b, k), we present an explicit mechanism that, using collaborative data only, enhances the diversity of recommendations made by (conversational) collaborative recommender systems.

We have experimented with four definitions of distance, for use when computing relative diversity. We found, perhaps counter-intuitively, that approaches based on Hamming Distance work better than those based on Inverse Pearson correlation.

Conversational collaborative recommenders are a new line of research, and enhancing the diversity of their recommendations is a new departure too. Future work could include: seeking better-motivated ways of boosting similarity; and more systematic investigation of good values for α, b and k. It would also be interesting to compare content-based approaches

to item distance with the approaches that we have reported in this paper, which use purely collaborative data.

We would also like to investigate the role of diversity over the course of the dialogue. In content-based recommenders, diversity has been found to be helpful in early cycles, when the user is exploring the space and making her short-term interests known; but in later cycles, when the user is homing in on a suitable item, diversity may be less appropriate (McGinty and Smyth 2003). We would like to find out whether this applies to conversational collaborative recommenders too.

The success rates and the dialogue lengths for the conversational systems (whether diversity-enhanced or not) that we found in the experiments would not be acceptable in practice. An essential avenue of future research would be to investigate these aspects of performance under less severe experimental conditions. Replacing artificial users by real users is one way to do this.

In any case, we believe that our ideas for enhancing diversity can be used even in single-shot collaborative recommenders. In single-shot systems, diverse results are shown, not so much to improve the likelihood of useful feedback, but rather to show the user meaningful alternative items, especially when screen estate is limited (Smyth and McClave 2001).

Acknowledgements This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/136. We are grateful to Professor Barry Smyth and the anonymous referees for their advice and to the GroupLens project team for making their data available: `www.grouplens.org`.

References

Balabanović M, Shoham Y (1997) Fab: content-based, collaborative recommendation. Communications of the ACM 40(3):66–72

Bradley K, Smyth B (2001) Improving recommendation diversity. In: O'Donoghue D (ed) Proceedings of the 12th Irish conference on artificial intelligence and cognitive science. NUI Maynooth, pp 85–94

Bridge D, Ferguson A (2002) Diverse product recommendations using an expressive language for case retrieval. In: Craw S, Preece A (eds) Proceedings of the 6th European conference on case-based reasoning. Springer, pp 43–57

Bridge D, Göker MH, McGinty L, Smyth B (2006) Case-based recommender systems. Knowledge Engineering Review 20(3):315–320

Bridge D, Kelly JP (2005) Diversity-enhanced conversational collaborative recommendations. In: Creaney N (ed) Proceedings of the 16th Irish conference on artificial intelligence & cognitive science. University of Ulster, pp 29–38

Burke RD, Hammond KJ, Young BC (1997) The findme approach to assisted browsing. IEEE Expert 12(5): 32–40

Doyle M, Cunningham P (2000) A dynamic approach to reducing dialog in on-line decision guides. In: Blanzieri E, Portinale L (eds) Proceedings of the 5th European workshop on case-based reasoning. Trento, Italy, Springer: pp 49–60

Herlocker J, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22(1):5–53

Herlocker JL (2000) Understanding and improving automated collaborative filtering systems. Ph.D. thesis, University of Minnesota

Linden, G, Hanks S, Lesh N (1997) Interactive assessment of user preference models: the automated travel assistant. In: Jameson A, Paris C, Tasso C (eds) Proceedings of the 6th international conference on user modeling. Springer, pp 67–78

McGinty L, Smyth B (2002) Comparison-based recommendation. In: Craw S, Preece A (eds) Proceedings of the 6th European conference on case-based reasoning. Aberdeen, Scotland, Springer, pp 575–589

McGinty L, Smyth B (2003) On the role of diversity in conversational recommender systems. In: Ashley K, Bridge D (eds) Proceedings of the 5th international conference on case-based reasoning. Springer, pp 276–290

McSherry D (2002) Diversity-conscious retrieval. In: Craw S, Preece A (eds) Proceedings of the 6th European conference on case-based reasoning. Springer, pp 219–233

Pu P, Viappiani P, Faltings B (2006) Increasing user decision accuracy using suggestions. In: Grinter R et al (eds) Proceedings of the conference on human factors in computing systems (CHI). ACM Press, pp 121–130

Rafter R, Smyth B (2004) Towards conversational collaborative filtering. In: McGinty L, Crean B (eds) Proceedings of the 15th artificial intelligence and cognitive science conference. pp 147–156

Reilly J, McCarthy K, McGinty L, Smyth B (2004) Dynamic critiquing. In: Funk P, González-Calero PA (eds) Proceedings of the 7th European conference on case-based reasoning. Madrid, Spain, Springer, pp 763–777

Resnick P, Varian HR (1997) Recommender systems. Communications of the ACM 40(3):56–58

Riedl J, Konstan J (2002) Word of mouse: the marketing power of collaborative filtering. Warner Books

Sarwar BM, Karypis G, Konstan JA, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international world Wide Web Conference. pp 285–295

Schmitt S (2002) simVar; A similarity-influenced question selection criterion for e-sales dialogs. Artificial Intelligence Review 18(3–4):195–221

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

Smyth B, McClave P (2001) Similarity vs. diversity. In: Aha DW, Watson I (eds) Proceedings of the 4th international conference on case-based reasoning. Springer, pp 347–361

Ziegler C-N, McNee SM, Konstan JA, Lausen G (2005) Improving recommendation lists through topic diversification. In: Proceedings of the 14th international World Wide Web Conference. ACM Press, pp 22–32