A Case-Based Solution to the Cold-Start Problem in Group Recommenders

Lara Quijano-Sánchez
Universidad Complutense de Madrid, Spain
lara.quijano@fdi.ucm.es

Derek Bridge
University College Cork, Ireland
d.bridge@cs.ucc.ie

Belén Díaz-Agudo and **Juan A. Recio-García**
Universidad Complutense de Madrid, Spain
belend@sip.ucm.es, jareciog@fdi.ucm.es

Abstract

In this paper we offer a potential solution to the cold-start problem in group recommender systems. To do so, we use information about previous group recommendation events and copy ratings from a user who played a similar role in some previous group event. We show that copying in this way, i.e. conditioned on groups, is superior to copying nothing and also superior to copying ratings from the most similar user known to the system.

1 Introduction

Groups often dine in restaurants together; visit historic sights, galleries and museums together; attend concerts, the theatre and the cinema together; vacation together; cook and eat together; watch TV together. They must select the items which they intend to consume together, ranging from restaurants to TV programmes, in a way that reconciles the different preferences and personalities of the group members. For this, they may seek the support of a recommender system. But where the majority of recommender systems suggest items based on the preferences of an individual consumer, *group recommender systems* suggest items taking into account the preferences and personalities of the members of a group [4].

In this paper, in the context of movie recommendation to groups of friends, we consider a group recommender system that aggregates the results of running a *single-person recommender system* to predict movie ratings for each member of the group. The single-person recommender that we use is a *user-based collaborative recommender system* [3], which predicts a user's rating for a candidate movie from the ratings given to that movie by a neighbourhood of users who are similar to that user.

But collaborative recommenders suffer from *cold-start problems* [3; 14]. In particular, a user-based collaborative recommender finds it difficult to make good predictions for new users, for whom it has few ratings. The group recommender inherits this problem too because it aggregates the predicted ratings from the single-user collaborative recommender. Solutions to the cold-start problem for single-person recommenders are summarized in [14] and include: non-personalized recommendations for cold-start users using population averages; intelligent ways to solicit more ratings (e.g. [2; 12]); and hybrid recommenders that resort to content-based recommendations when there are insufficient ratings to make collaborative recommendations (e.g. [1; 6]).

The contribution of this paper is to introduce and evaluate a *case-based reasoning* (CBR) solution to this problem. We use a case base in which each case records a previous group movie recommendation event. When a group requests a new recommendation but where one or more of the group members is in cold-start, we find a case that describes a previous recommendation event where there are users who are not in cold-start but who play similar roles in their group to the roles the cold-start users play in the active group. We temporarily copy ratings from the users in the case to the corresponding users in the active group and only then proceed to run the single-person recommender and to aggregate its results. It is natural to use a CBR approach because, similar events recur: the same group (perhaps with some small variations) repeats activities together; and some age, gender and personality distributions will tend to recur too (e.g. two adults with two children going to the movies).

CBR has been used in recommender systems before (e.g. [13]) and explicit parallels between CBR and user-based collaborative recommenders have been drawn (e.g. [7]). But we are unaware of any previous use of CBR in group recommenders or in solutions to the cold-start problem.

Section 2 describes our group recommender method; Section 3 describes how we have extended our method to solve the cold-start problem; Section 4 proposes systems against which our system can be compared; Section 5 describes the dataset that we have used in our experiments; Section 6 presents our experimental method; Section 7 contains results; and Section 8 concludes our work.

2 Social Group Recommender Systems

Suppose there are n users, $U = \{u : 1 \ldots n\}$, and m items (e.g. movies), $I = \{i : 1 \ldots m\}$. Let $r_{u,i}$ be the rating that user u assigns to item i. Ratings are on a numeric scale, e.g. 1 = terrible and 5 = excellent, but $r_{u,i} = \perp$ signals that u has not yet rated i. Let $G_a \subseteq U$ be an active group of users, in our case a group who intend going to see a movie together. The goal is to recommend to G_a a set of k items drawn from a set of candidate target items $T_a \subseteq I$. For example, T_a could be the set of movies showing this week at the local multiplex.

If none of the users in G_a is in cold-start the group recommender system will work asfollows:

Step 1: First, it uses a user-based collaborative recommender to predict a rating $\hat{r}_{u_a,i}$ for each $u_a \in G_a$ for the various i in T_a. The user-based collaborative recommender that we use works as described in [3; 14]. In brief, it computes the similarity between u_a and each other user $u \neq u_a$ who has rated i; it retrieves u_a's 20 nearest neighbours, i.e. the 20 users who are most similar to u_a; and its prediction $\hat{r}_{u_a,i}$ is then a weighted average of the neighbours' actual ratings for i.

Step 2: Second, for each i, it aggregates the predicted ratings of each $u_a \in G_a$ to give a predicted group rating for that item, i.e.

$$\hat{r}_{G_a,i} \hat{=} \underset{u_a \in G_a}{\mathrm{F}} \hat{r}_{u_a,i} \tag{1}$$

where F is the aggregation function, which we discuss in more detail below.

Step 3: Finally, it recommends the k items $i \in T_a$ for which the predicted group ratings $\hat{r}_{u_a,i}$ are highest.

Possible aggregation functions F include *least misery* (taking the minimum) and *most pleasure* (taking the maximum) [5]. We experimented with both before [8], and we found *most pleasure* to give better results, and so we adopt that here.

However, our previous work showed an improvement in the accuracy of predicted group ratings by taking into account the *personality* of the users in the group and the strength of their connections, which we refer to as their *trust* [10; 8; 11]. We refer to our recommender that takes this extra social information into account as being *social* and the method it uses as being *delegation-based*. Specifically then, we have:

$$\hat{r}_{G_a,i} \hat{=} \max_{u_a \in G_a} \mathrm{dbr}(\hat{r}_{u_a,i}, G_a) \tag{2}$$

Here the *most pleasure* principle (maximum) is not applied directly to individual predicted ratings, $\hat{r}_{u_a,i}$. The ratings are modified by the dbr function, which takes into account personality and trust values within the group G_a.

We obtain the personality of each user u by requiring group members complete a personality test on registration with the recommender. The details of the personality test are in [16]. In a real application, such as the Facebook social group recommender that we have built [9], trust between users two users can be based on distance in the social network, the number of friends in common, relationship duration, and so on.

Space limitations preclude a detailed presentation of dbr but it is defined in, for example, [10].

We will designate the social group recommender system that we have outlined in this section by *Soc*.

3 Using CBR for Cold-Start Users

As we have explained, an active user with few ratings is said to be in cold-start. The problem that this causes for the kind of recommenders that we have been discussing is that it becomes difficult to find a reliable neighbourhood of similar users from which predictions can be made. One solution is to temporarily copy some ratings into the profile of the active cold-start user from a similar user who has additional ratings. Similarity in this case would be measured using demographic information [14] because the active user has insufficient ratings to find a similar user based on co-rated items.

A group recommender can take the same approach when members of the group are in cold-start: prior to predicting individual ratings, it can temporarily augment the ratings profiles of group members who are in cold-start with ratings that are copied from the profiles of similar users. But in a group recommender, we can go further than using just demographic information. In our work, we investigate how to reuse ratings from similar users in similar groups in a case-based fashion.

Assume a case base CB in which each case $c \in CB$ records a previous group movie recommendation event. Each case will have the following structure:

$$\langle id_c, \langle G_c, T_c \rangle, i_c \rangle$$

where id_c is a case identification number. The *problem description* part of the case comprises:

- $G_c \subseteq U$, the group of users who used the recommender previously. For each user $u \in G_c$, we will know u's age and gender; u's ratings, $r_{u,i}$, for some set of items; and u's personality value. For each pair of users $u \in G_c, v \in G_c, u \neq v$, we will know the trust value.
- $T_c \subseteq I$, the set of items that the users were choosing between. In our case, these were the movies that were at the local multiplex on the occasion when this group used the recommender.

And the *solution* part of the case contains just $i_c \in T_c$, the item that the group agreed on. In our case, this is the movie that the group went to see together.

If none of the users in G_a is in cold-start, then the system will work either in the fashion described in Section 2.

But suppose, on the other hand, that one or more members of G_a are in cold-start. We define this simply using a threshold, θ: a user u_a is in cold-start if and only if the number of items s/he has rated is less than θ. In this case, we need to use the CBR. For each user who is in cold-start, we will copy ratings from the *most similar user in the most similar group* in the case base, as follows.

Case retrieval

We can write the problem statement as $PS = \langle G_a, T_a \rangle$. We will find the *most similar case*, c^*, in the case base:

$$c^* \hat{=} \arg\max_{c \in CB} \mathrm{sim}(PS, c) \tag{3}$$

The similarity between a problem statement $PS = \langle G_a, T_a \rangle$ and a case $c = \langle id_c, \langle G_c, T_c \rangle, i_c \rangle \in CB$, $\mathrm{sim}(PS, c)$, is calculated on the basis of group similarity:

$$\mathrm{sim}(\langle G_a, T_a \rangle, \langle id_c, \langle G_c, T_c \rangle, i_c \rangle) \hat{=} \mathrm{gsim}(G_a, G_c) \tag{4}$$

This means that in our work case similarity takes only the groups, G_a and G_c, into account; it does not take into account the items, T_a and T_c. T_c contains the items that G_c contemplated in the past, but T_a contains items that G_a is contemplating right now, e.g. movies that have just come to town, and these sets need not even overlap.

This process requires a definition of group similarity, gsim. We compute the similarity of any pair of groups, G and G', from the similarity of the users in the two groups, $\mathrm{psim}_{CB}(u, G, u', G'), u \in G, u' \in G'$. Specifically, we define $\mathrm{gsim}(G, G')$ to be the average similarity of each user u in G to his/her most similar user in G'. Note that we do not prevent two or more people from G being associated with the same user $u' \in G'$ (and vice versa). This fact allows us to easily compare groups of different sizes. It does mean that, if two or more users from G_a are in cold-start, they may all copy ratings from the same user $u' \in G$.

We define $\text{psim}_{CB}(u, G, u', G')$, the similarity between two users in groups, as an average similarity over the data that we hold about them: their ratings, gender, ages, personality values and trust values. Details are in [11].

Case reuse

Next, for each user u_a in G_a who is in cold-start, we find the *most similar user* u^* in case c^* who has rated movies that u_a has not. Let G^* be the group of people described in case c^*:

$$u^* \hat{=} \underset{u \in G^* \wedge \exists i, r_{u_a,i} = \bot \wedge r_{u,i} \neq \bot}{\arg\max} \text{psim}_{CB}(u_a, G_a, u, G^*) \quad (5)$$

In the case of more than one such user, we choose the one from whom we can copy the most ratings, i.e. the one who has most ratings for movies that u_a has not rated. Then, temporarily (for the purposes of making u_a's prediction for the items in T_a), we copy into u_a's profile the rating for each item i that u^* has rated ($r_{u^*,i} \neq \bot$) that u_a has not ($r_{u,i} = \bot$).

With each cold-start user's profile augmented in this way, we can then proceed to compute group recommendations in the fashion described in Section 2. But, it should now be less problematic finding neighbourhoods for the users who are in cold-start because they now have augmented user profiles. We will designate this system by *Soc-CB*.

4 Other Recommenders for Cold-Start Users

An obvious question is whether it makes a difference that our case-based solution to the cold-start problem in group recommenders works on a group basis at all. Why copy ratings from the most similar user in the most similar group? Why not copy ratings simply from the most similar user in the case base as a whole? Or why not copy ratings from the most similar user known to the system? Systems that work in these different ways will be useful for comparisons in our experiments, hence we define both of these more precisely now.

Consider the set of users who appear in at least one case in the case base:

$$U_{CB} \hat{=} \{u : \exists c = \langle id_c, \langle G_c, T_c \rangle, i_c \rangle \in CB \wedge u \in G_c\} \quad (6)$$

When trying to predict group G_a's rating for an item $i \in T_a$, then for any user $u \in G_a$ who is in cold-start, we could find, and copy ratings from, the most similar user in U_{CB}:

$$u^* \hat{=} \underset{u \in U_{CB} \wedge \exists i, r_{u_a,i} = \bot \wedge r_{u,i} \neq \bot}{\arg\max} \text{psim}_{U_{CB}}(u_a, u) \quad (7)$$

This is different from first finding the most similar group and then, for each active user in cold-start, copying ratings from the most similar user in that group. Our case-based approach is conditioned on the groups; this alternative is not.

We will designate this recommender by *Soc-UCB*.

The second of our two alternative cold-start recommenders ignores the case base altogether. It simply finds, and copies ratings from, the most similar user in U (the entire set of users), wholly ignoring whether they have previously participated in group recommendations or not. Hence,

$$u^* \hat{=} \underset{u \in U \wedge \exists i, r_{u_a,i} = \bot \wedge r_{u,i} \neq \bot}{\arg\max} \text{psim}_{U}(u_a, u) \quad (8)$$

We will designate this recommender by by *Soc-U*.

5 Group Recommender Dataset

We need a dataset with which we can evaluate our case-based solution to the cold-start problem in group recommenders. We are not aware of a public dataset for group recommenders, hence we created our own.

We started from the MovieLens 1M dataset (`www.grouplens.org`). It gives us around 1 million ratings on a scale of 1 to 5 for around 6040 users for nearly 4000 movies. For each user, it records gender, age range, and at least twenty ratings. We impute a personality value to each used based on the population norms in [15].

We created 100 groups. Group members are chosen at random from all users in the MovieLens dataset but subject to the following restrictions: in a group, users are distinct (but a user may be in more than one group); all users are in the same age range; and we ensure that there are at least 15 movies which are co-rated by all members of the group. When we create cases, these 15 movies will be the set T_c. Their ratings are withheld from the recommender, because it would not in general know a user's actual ratings for the candidate movies.

We conducted a Facebook poll in which we asked respondents to tell us, for the last five times that they went to the cinema in a group, how large the group was. We used the frequencies to create our groups. We have 50 groups of size 2, 18 of size 3, 16 of size 4, 7 of size 5, 5 of size 6, and 4 where we took the size to be 7.

As we have discussed, in our Facebook application, trust is computed from Facebook data (distance in the social network, etc.), but that is not available to us for the users in the MovieLens dataset. Rather than simply imputing trust values at random, we chose to base them on the degree of shared taste as revealed by co-rated items.

To create a case, we need to indicate which of the 15 movies in T_c the group will actually have chosen. But we cannot ask random groups of MovieLens users to work out which of their 15 candidate movies they would have gone to see together. We used four human 'experts' who were given all the information about a group's members G_c and the candidate movies T_c (including the actual ratings by the members of G_c for the items in T_c) and were asked to decide which of the movies the group would be most likely to settle on. Each expert evaluated 50 cases, hence each of the 100 groups was evaluated by two experts (not always the same two). Experts were asked to give an ordered list of the three movies from T_c that they thought the members of G_c would agree on, and we combined the experts' judgements into a single final ordered list. We will designate this ordered list by E (for 'Expert') and we will use E_1 to mean movies in the first position in E, E_2 to mean movies in the first and second positions in E, and so on.

6 Evaluation Methodology

The dataset that we have created has 100 movie-going events. We use a leave-one-out cross-validation methodology, where we remove each case in turn from the case base and present it to the recommenders. We compare their recommendations with the experts' judgements.

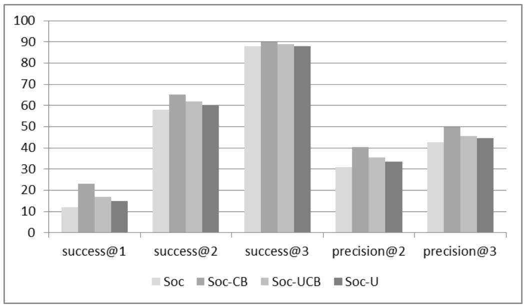

Figure 1: Results for $\theta = 20$

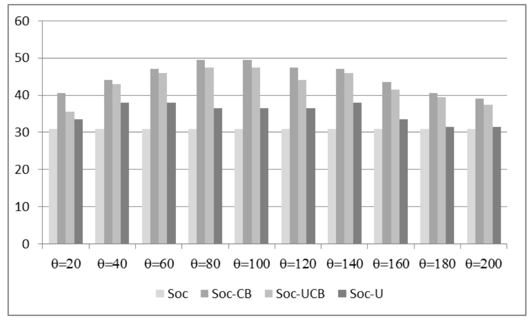

Figure 2: Results for $precision@2$

We report results from four recommenders: *Soc*, *Soc-CB*, *Soc-UCB*, and *Soc-U*. Each recommender recommends the top $k = 3$ movies from the 15 candidates. Let R be the set of recommendations made by a particular recommender. Then we want to compare R with E from above. We computed total $success@n$ for $n = 1, 2, 3$, where $success@n$ = 1 if $\exists i, i \in R \wedge i \in E_n$ and is 0 otherwise. For example, when using $success@2$, we score 1 each time there is at least one recommended movie in the top two positions of E. We also computed total $precision@n$ for $n = 1, 2, 3$, where $precision@n \hat{=} |\{i : i \in R \wedge i \in E_n\}|/n$. For example, if no recommended movie is in the top two positions in E, then $precision@2 = 0$; if one recommended movie is in the top two positions in E, then $precision@2 = 0.5$.

We repeat the experiments with different cold-start thresholds (θ). For $\theta = 20$, just over ten users are in cold-start; with $\theta = 40$, an additional twenty users are in cold-start; and then as θ goes up by 20, the number of users in cold-start grows by about an additional ten each time. (The threshold excludes the 15 ratings for T_a withheld from the recommender.)

7 Results

Figure 1 shows $success@n$ for $n = 1, 2, 3$ and $precision@n$ for $n = 2, 3$ ($precision@1 = success@1$ and is therefore not shown) for cold-start threshold $\theta = 20$.

Results show that as n gets bigger, results improve but differences between systems become less pronounced: with bigger n it is simply easier to make a recommendation that matches an expert judgement. The most important observation is that the *Soc-CB* system out-performs the *Soc-UCB* system, which out-performs the *Soc-U* system, which out-performs the *Soc* system. So, a cold-start strategy that is conditioned on groups copies ratings in a more informed and successful way than strategies that copy without regard to groups, and copying ratings is more successful than having no cold-start solution.

We tried out a similar cold-start solution in the context of a single-person recommender, where a single active user seeks movie recommendations. If the active user was in cold-start, we copied ratings from a similar user in U. Interestingly, doing so made no or almost no change to the $success@n$ and $precision@n$ results (not shown here) across several definitions of similarity. We conclude that, for our movie data, conditioning on groups really does seem to be the most effective way to use this cold-start solution.

We have also studied the impact of varying θ (from 20 to 200), Figure 2. In other words, more and more users are regarded as being in cold-start and are given ratings from other users. The results for *Soc* itself remain the same for all values of θ because this system has no cold-start strategy. For the other systems, we see that results improve and then fall off as θ increases. For example, for *Soc-CB*, results improve until $\theta = 100$. For this system, 100 is the cut-off point: users with fewer than 100 ratings are ones we should regard as being in cold-start. A higher threshold treats so many users as being in cold-start that the tastes of the active group are swamped by the ratings copied from other users, causing system performance to decrease. The graph is for $precision@2$ but we observed the same pattern of results for all other measures.

8 Conclusions

We have presented a new solution to the cold-start problem in a collaborative group recommender. We use a case base of group recommendation events and copy ratings into the profile of users who are in cold-start from their most similar user in the most similar group in the case base. Our experiments on movie data show that, for users with fewer than 100 ratings, this strategy improves the quality of the group recommendations.

A side-product of the research has been the construction of a dataset for group recommender research. We recognize that it has limitations: for example, it contains no family groups (e.g. parents with children) since members of a group are selected to be in the same age range; and its imputation of personality and trust values is too simplistic.

There is much that can be done to take this work forward. For us, the next step is to consider a case base in which we more explicitly arrange that there be cases (e.g. movie-going events) that involve groups whose members have a high degree of overlap with the members of the active group, so that we can experiment with the situation where the same group (or nearly the same group) consumes items together on a frequent basis. We also intend to consider richer case representations to take into account such things as timestamps, predicted and actual ratings from group members, and the dynamics of reaching a consensus (e.g. changes in group membership and changes in the selected item). We hope too to gather more data from our Facebook application and use this data to overcome the limitations of our current dataset.

References

[1] M. Balabanovic and Y. Shoham. Fab: Content-based, collaborative recommendation. *Comms. of the Association of Computing Machinery*, 40(3):66–72, 1997.

[2] N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Fourth ACM International Conference on Web Search and Data Mining*, 2011.

[3] J. L. Herlocker. *Understanding and Improving Automated Collaborative Filtering Systems*. PhD thesis, University of Minnesota, 2000.

[4] A. Jameson and B. Smyth. Recommendation to groups. In *The Adaptive Web, Methods and Strategies of Web Personalization*, Lecture Notes in Computer Science, 4321, pages 596–627. Springer, 2007.

[5] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, 2004.

[6] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Procs. of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 187–192, 2002.

[7] D. O'Sullivan, D. C. Wilson, and B. Smyth. Improving case-based recommendation: A collaborative filtering approach. In S. Craw and A. Preece, editors, *Procs. of the 6th European Conference on Case-Based Reasoning*, pages 278–291. Springer, 2002.

[8] L. Quijano-Sánchez, J. A. Recio-García, and B. Díaz-Agudo. An architecture and functional description to integrate social behaviour knowledge into group recommender systems. In *Applied Intelligence*. in press, 2013.

[9] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz. Happy movie: A group recommender application in Facebook. In *24th International Florida Artificial Intelligence Research Society Conference, FLAIRS- 2011*, 2011.

[10] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz. Social factors in group recommender systems. In *ACM-TIST, TIST-2011-01-0013*. in press, 2011.

[11] Lara Quijano-Sánchez, Derek Bridge, Belén Díaz-Agudo, and Juan A. Recio-García. A case-based solution to the cold-start problem in group recommenders. In Belén Díaz-Agudo and Ian Watson, editors, *Case-Based Reasoning Research and Development (Procs. of the 20th International Conference on Case-Based Reasoning)*, LNAI 7466, pages 342–356. Springer, 2012.

[12] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 2002 International Conference on Intelligent User Interfaces*, pages 127–134, 2002.

[13] F. Ricci, B. Arslan, N. Mirzadeh, and A. Venturini. ITR: A case-based travel advisory system. In S. Craw and A. Preece, editors, *Procs. of the 6th European Conference on Case-Based Reasoning*, pages 613–641. Springer, 2002.

[14] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web, Methods and Strategies of Web Personalization*, Lecture Notes in Computer Science 4321, pages 291–324. Springer, 2007.

[15] N. A. Schaubhut. *Technical Brief for the THOMAS-KILMANN CONFLICT MODE INSTRUMENT*. CPP Research Department, 2007.

[16] K. W. Thomas and R. H. Kilmann. *Thomas-Kilmann Conflict Mode Instrument*. Tuxedo, N.Y., 1974.