A Group Recommender for Movies Based on Content Similarity and Popularity

Maria S. Pera* and Yiu-Kai Ng

Computer Science Department
Brigham Young University
Provo, Utah 84602, U.S.A.
Email: mpera@cs.byu.edu, ng@compsci.byu.edu

Abstract

People are gregarious by nature, which explains why group activities, from colleagues sharing a meal to friends attending a book club event together, are the social norm. Online group recommenders identify items of interest, such as restaurants, movies, and books, that satisfy the collective needs of a group (rather than the interests of individual group members). With a number of new movies being released every week, online recommenders play a significant role in suggesting movies for family members or groups of friends/people to watch, either at home or at movie theaters. Making group recommendations relevant to the joint interests of a group, however, is not a trivial task due to the diversity in preferences among group members. To address this issue, we introduce *GroupReM* which makes movie recommendations appealing (to a certain degree) to members of a group by (i) employing a *merging strategy* to explore individual group members' interests in movies and create a profile that reflects the preferences of the group on movies, (ii) using *word-correlation factors* to find movies similar in content, and (iii) considering the *popularity* of movies at a movie website. Unlike existing group recommenders based on collaborative filtering (CF) which consider ratings of movies to perform the recommendation task, GroupReM primarily employs (personal) *tags* for capturing the contents of movies considered for recommendation and group members' interests. The design of GroupReM, which is simple and domain-independent, can easily be extended to make group recommendations on items other than movies. Empirical studies conducted using more than 3,000 groups of different users in the MovieLens dataset, which are various in terms of numbers and preferences in movies, show that GroupReM is highly *effective* and *efficient* in recommending movies appealing to a group. Experimental results also verify that GroupReM outperforms popular CF-based recommenders in making group recommendations.

Keywords: Group recommender, content-similarity, popularity, movie

*Corresponding author: Maria S. Pera, mpera@cs.byu.edu, 801-422-5815, 3361 TMCB, Computer Science Department, Brigham Young University, Provo, Utah, 84602, USA

1 Introduction

During the past decades, a number of recommender systems have been developed to aid individual users in finding items of interest among the millions available, which include songs [SWW10], books [PN11], and websites [CBV10], to name a few. These recommenders, however, are tailored only to the needs of individual users. As people are gregarious by nature, a variety of activities involve groups of people who participate either online or in an old-fashioned manner, i.e., in person. To meet the demands of groups of users, group recommenders [AYRC+09, GXL+10, RGJDSRDA09] have been proposed to identify items, such as vacation packages [MSC+06], restaurants [PPC08], TV shows [Mas04], songs [CS11, MA98], or movies [RAYC+10, OCKR01], that appeal to a group as a whole (rather than individual users). As claimed by Gartrell et al. [GXL+10], effective group recommendations can have a positive impact on people's social activities. Suggesting items that satisfy (to a certain degree) the needs of members of a group, however, is a challenging task due to the diverse interests of group members, even more so when dealing with groups consisting of dissimilar members in terms of their preferences [AYRC+09, BMR10].

One of the in-demand recommendation tasks is to suggest movies to a group. Movies offer a popular group activity for friends, families, and colleagues who gather to either see a movie at the cinema or watch a DVD at home. These people often turn to experts' reviews to find movies that match their interests and/or reach a consensus on their own regarding the movies to watch. Group recommenders on movies can streamline this process by directly suggesting movies appealing to a group. While the majority of the recommenders that have recently been introduced to make group recommendations on movies are based on the popular collaborative filtering (CF) strategy [GXL+10, RAYC+10], to the best of our knowledge, none of them adopts the content-based strategy to exploit descriptive information on movies to perform the recommendation task. In this paper, we introduce GroupReM, a group recommender system on movies. Our proposed top-N recommender, which is based on a content-based strategy to identify a ranked set of N movies that best match the (content of movies of) interest to a group, differs from existing CF-based group recommenders that adopt various strategies for predicting (individual/group) ratings on movies and suggest the movies with the highest overall rating to a group [YLAY09]. These recommenders are restricted, since "similar-minded" individuals at a movie website and the existence of large historical data to guarantee rating overlap among users [ORTN10] are required to make recommendations. GroupReM, on the other hand, simply relies on data readily available on social websites, which are tags and their frequencies of occurrence, along with bookmarked movies, to suggest movies to a group.

GroupReM considers semantic information of movies, i.e., (personal) tags at a movie website, to capture both the (i) content of movies and (ii) the preferences of members of a given group G on movies archived at the website. GroupReM applies a *rank aggregation* model on two different measures, *group appealing* and *global popularity*, computed for each candidate movie M to be considered for recommendation. The former captures the content similarity between M and the group profile of G, whereas the latter reflects the popularity of M at the movie website. GroupReM anticipates that popular movies, which are frequently bookmarked, that are similar in content (based on tags) to the group profile, which characterizes G, are of interest to the members of G. In matching (the tags in)

M and the profile of G, GroupReM does not impose an exact-match constraint. Instead, GroupReM relies on pre-computed word-correlation factors [KN06] to determine inexact, but analogous, tags in M and G, in addition to exact-matched tags, to more accurately capture the degree of appealing of M to G.

GroupReM is (i) *simple*, since it solely employs a standard measure to combine the aforementioned content-similarity and popularity scores, (ii) *fast*, since it takes on the average less than a second to make recommendations for a group (of up to eight members), and (iii) *scalable*, since GroupReM can identify movies that capture the common interests of a group regardless of its *size* and the *degree of cohesiveness* among group members. Moreover, GroupReM requires neither training nor domain-specific knowledge to select movies to be recommended and thus can directly be adopted to make recommendations on items other than movies. We have conducted an empirical study using more than 3,000 groups of users from the MovieLens dataset [aUoM] and verified that GroupReM (i) generates relevant recommendations on movies tailored to the needs of a group and (ii) significantly outperforms CF-based group recommenders.

The remaining of this paper is organized as follows. In Section 2, we discuss existing group recommenders and compare their recommendation strategies with GroupReM. In Section 3, we detail the design of GroupReM. In Section 4, we present the empirical study conducted to assess (compare, respectively) the performance of GroupReM (GroupReM with existing CF-based group recommenders, respectively) and illustrate the effectiveness and efficiency of GroupReM. In Section 5, we give a concluding remark and directions for future work.

2 Related Work

In this section, we present a number of existing group recommenders that suggest different types of items, including vacations [MSC+06], recipes [BF10], TV programming [BCC+09], and music [CBH02], and compare their recommendation approaches with GroupReM. Thereafter, we introduce representative work on recently-developed group recommenders on movies [GXL+10, RAYC+10] which differ from GroupReM in their design methodologies. An in-depth discussion on group recommenders can be found in [BC11, JS07].

2.1 Non-Movie Group Recommenders

As defined in [BF10], there are two strategies commonly-adopted for generating group recommendations: the *aggregated models* and *aggregated predictions*. The former combines individual user models, i.e., individual user profiles that capture the preferences of a group member, into a group model from where items to be recommended for the group are identified, whereas the latter generates predictions for individual group members and then aggregates the predictions to suggest items for the group. Empirical studies conducted and presented in [BF10] suggest that the aggregated models strategy (which is employed by GroupReM) generally outperforms the aggregated predictions strategy.

Flytrap [CBH02], which identifies musical tracks for a group, learns the music preferences of a user U based on the songs U has listened to and the numerical votes casted by

U for the songs. Flytrap considers (i) relationships among musical genres, (ii) the influence artists have on one another, and (iii) the transitions in between songs people tend to make to perform the recommendation task. The recommender relies on domain-specific information and thus cannot be extended to suggest items other than songs, contrary to GroupReM which can directly be employed for recommending non-movie items.

CATS (Collaborative Advisory Travel System) [MSC+06] assists groups of friends in planning skiing vacations. CATS relies on an incremental method which analyzes individual user's critiques on the proposed recommendations to refine the recommendations generated for the group. Unlike GroupReM, CATS depends on user feedbacks to narrow the search space, i.e., identify items that satisfy the need of an individual, as well as a group, which is a burden on the users. Moreover, CATS has been designed to recommend items to a group of at most four users, which is a limitation, as opposed to GroupReM which does not impose a constraint on the number of group members in performing its recommendation task.

Berkovsky and Freyne [BF10] recommend recipes to families through an eHealth portal. The proposed CF-based group recommender considers the (i) ratings assigned to recipes on the eHealth portal and (ii) weight, i.e., influence, of each individual group member computed according to his/her activities on the portal in making recommendations. Unlike the recommender introduced in [BF10], GroupReM relies on the semantic content and popularity of movies to accurately perform the recommendation task.

Cantador and Castells [CC11] introduce an ontology-based group recommendation strategy. The proposed approach identifies users that share similar tastes/preferences, i.e., "communities of interest", according to individuals' ontology-based profiles. These clusters of related users are then exploited to generate group profiles and perform the recommendation task. Similar to GroupReM, the strategy in [CC11] is primarily content-based. However, the approach presented in [CC11] is employed to suggests photos (instead of movies), relies on ontology concepts to determine the similarity among users/items (unlike GroupReM that depends on user-defined keywords, i.e., tags, to capture users' preferences and items' descriptions), and according to the authors "'more sophisticated and statistically significative experiments need to be performed in order to properly evaluate" the correctness of applying the clustering techniques presented in [CC11] for group modeling and content-based collaborative filtering recommendation.

Masthoff [Mas04] describes a number of recommendations strategies that merge individual user models in order to suggest TV shows that appeal to a group of users. Unlike GroupReM, (some of) the group recommendation strategies discussed in [Mas04] are inspired by Social Choice Theory. Boratto et al. [BCC+09] recommend TV programming to a group by first employing a hierarchical clustering algorithm using the cosine similarity metric, which determines the similarity among pairs of users, to identify a natural community G, i.e., a group. Thereafter, a profile for G is created, which reflects the overall preference of the members in G based on the average ratings given by members of G to programs. Based on the group profile, recommendations are generated. Unlike GroupReM which generates recommendations for groups regardless of the cohesiveness among group members, the group recommender in [BCC+09] makes recommendations for groups of similar-minded individuals only, which is a restriction, since in real life groups tend to include members that may not share similar interests in various TV programming.

2.2 Group Recommenders on Movies

A number of group recommenders that identify movies of interest to a group have been developed in the last few years, which include the systems introduced in [BMR10, GXL+10, OCKR01, RAYC+10]. Gartrell et al. [GXL+10] claim that some members of a group are more capable than others to influence the remaining group members in making decisions (i.e., relevant or non-relevant) on items suggested to the group. The authors consider several group factors, which include social interactions among group members, degrees of expertise of the members in the group, and dissimilarity among group members, to identify movies of interest to the group. While empirical studies conducted using ten groups have verified the effectiveness of the proposed recommender, it relies heavily on the interaction activities among group members that may not always exist or become available.

Baltrunas et al. [BMR10] conduct an empirical study to assess the effectiveness of alternative rank aggregation strategies, such as Spearman Footrule, Borda Count, Least Misery, and Average, for combining individual ranking predictions using a CF-based algorithm to make group recommendations. Similar to the approach in [BMR10], the group recommender developed by O'Connor et al. [OCKR01] adopts a Least Misery strategy to combine individual ratings predicted by a CF-based algorithm. Basu Roy et al. [RAYC+10] prune and merge rating lists predicted for individual members of a group G using the popular Average and Least Misery aggregation strategies, in addition to considering pairwise disagreement lists of movies, to recommend movies of interest to G. Unlike GroupReM, the group recommenders in [BMR10, OCKR01, RAYC+10] are based on the aggregated prediction strategy. According to the research work conducted in [BF10], this strategy has been empirically determined to be less effective than the aggregated model, which GroupReM adopts.

3 Our Proposed Group Recommender

In this section, we present our proposed recommender, GroupReM, which suggests movies appealing (to a certain degree) to members of a group who are users of a movie website, such as Netflix (netflix.com) and MovieLens (movielens.umn.edu). GroupReM relies on *tags* assigned to (represent the content of) movies and the *popularity* of each movie to make recommendations.

As group members of a movie website often have diverse preferences in movies, GroupReM first assesses the interest of each individual member U of a given group G based on the tags assigned by U to movies bookmarked in his/her profile. Tags and their frequencies of occurrence in group members' profiles are combined to create the *group profile* of G which reflects the common interests of the group members (as detailed in Section 3.1). Thereafter, using word-correlation factors (introduced in Section 3.2), GroupReM determines the movies, among the ones available at the website which are not included in the profile of any member of G, that are similar (based on tags) to the ones bookmarked by the members of G to a certain degree to generate the set of *candidate movies* that the group is likely interested in (as described in Section 3.3). Using a *rank aggregation function* (as presented in Section 3.4.3), GroupReM computes the overall ranking score of each candidate movie M. The *ranking score* of M is based on (i) the *group appealing* score of M for G (as

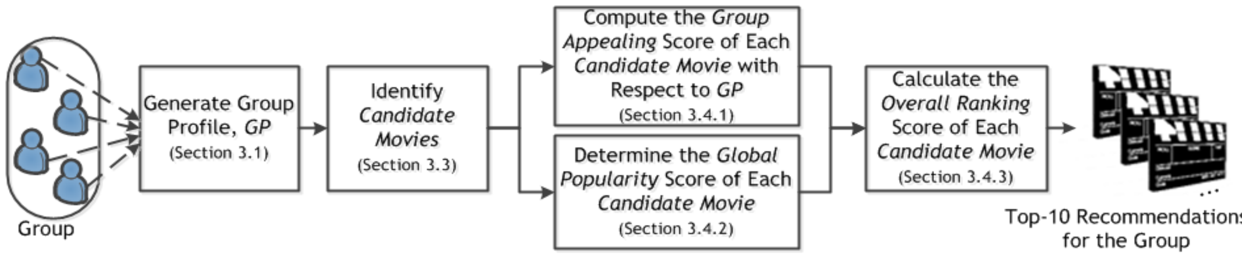


Figure 1: Processing steps of the proposed group recommender on movies, GroupReM

defined in Section 3.4.1) and (ii) the *popularity* score of M (as computed in Section 3.4.2). The former is calculated according to the number of tags assigned to (represent the content of) M that exactly-match or are analogous to the ones which characterize the profile of G, whereas the latter reflects the overall interest of the website users on M. The top-10 ranked candidate movies are recommended to G. The overall process of GroupReM is illustrated in Figure 1.

3.1 Creating a Group Profile

As the goal of group recommenders is to suggest movies of interest to a group, GroupReM analyzes the preference of each group member in movies and creates a *group profile* which reflects the types of movies preferred (to a certain degree) by the group as a whole. To construct the profile, GroupReM employs an *aggregated model* [BF10] that merges *individual user models*, i.e., the movies each group member is interested in, into a *group model*, which indicates the collective interest of the group members in movies.

GroupReM identifies the preference in movies of each individual member U of a group G by considering the movies bookmarked by U and tags assigned by U to the movies. Personal tags, i.e., tags defined by an individual user, are employed to represent (the content of) a movie M of interest to a user, as opposed to tags in the tag cloud[1] of M at a movie website, since GroupReM aims to capture U's description of M. Hereafter, GroupReM proceeds to create the *group profile* for G which includes all the personal tags (and their combined frequencies) assigned by the members of G to movies in their individual profiles. The *higher* the frequency of a tag T in G is, the more *adequately* T is in reflecting the *joint interest* of the group members on movies, since the high frequency of T reflects that T is more often used by members of G to describe movies they are interested in than other tags with lower frequencies.

Example 1 Consider a group G with three different MovieLens members. Figure 2 shows (a portion of) the profile of each member U in G, which includes the personal tags assigned by U to movies bookmarked in his/her profile. By combining the tags (and cumulating the corresponding frequencies) in the individual group member profiles, GroupReM creates a group profile for G. As shown in Figure 2, tags such as "drama" and "animation" reflect the types of movies that are of interest to each member of G, since the tags are included

[1] The tag cloud of a movie M, which provides the collective description on the content of M bookmarked by users at a movie website W, can be inferred by collecting each tag assigned to M by users at W, in addition to their $frequencies$.

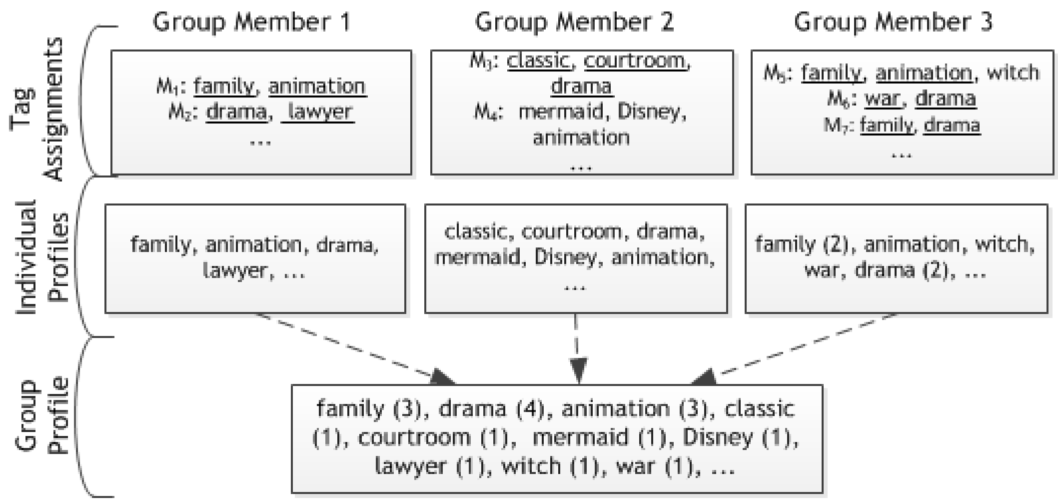


Figure 2: The group profile created by GroupReM for a given group G based on the tags in the profiles of three MovieLens users, who are members of G

in the personal profile of each group member, as opposed to tags such as "mermaid" and "war", which are preferred by one out of three group members. □

3.2 Word-Correlation Factors

GroupReM relies on the pre-computed word-correlation factors in the word-correlation matrix [KN06] to determine the *similarity* between any two tags, which facilitates the task of identifying *candidate movies* to be considered for recommendation (as detailed in Section 3.3). Moreover, GroupReM takes advantage of the word-correlation factors in calculating the *group appealing* score of a candidate movie with respect to a group profile (as discussed in Section 3.4.1).

Word-correlation factors were calculated using a set of approximately 880,000 Wikipedia documents (wikipedia.org). Each correlation factor indicates the degree of similarity of the two corresponding words[2] based on their (i) *frequency of co-occurrence* and (ii) *relative distances* in each Wikipedia document. Wikipedia documents were chosen for constructing the word-correlation matrix, since they were written by more than 89,000 authors with different writing styles, and the documents cover a wide range of topics with diverse word usage and contents. Compared with synonyms/related words compiled by the well-known WordNet (wordnet.princeton.edu) in which pairs of words are not assigned similarity weights, word-correlation factors provide a more sophisticated measure of word similarity. Despite the existence of a number of measures that rely on WordNet to determine the semantic similarity between pairs of words, such as Lesk [BT03] and LCH [LC98], GroupReM depends on word-correlations, which have been successfully adopted to determine the similarity between words in a number of applications, such as document classification [PN10] and text

[2]Words in the Wikipedia documents were *stemmed* (i.e., reduced to their grammatical roots) after all the *stopwords*, such as articles, conjunctions, and prepositions, which do not play a significant role in representing the content of a document, were removed. From now on, unless stated otherwise, (key)words/tags refer to non-stop, stemmed (key)words/tags.

retrieval [QPN12].

3.3 Identifying Candidate Movies to be Recommended

As the number of movies available at a movie website W can be large, i.e., in the hundreds of thousands, it is inefficient to analyze each movie of W to identify those of interest to the members of a group G at W, since the comparisons would significantly prolong the processing time of GroupReM to make recommendations. To minimize the number of comparisons and thus reduce the processing time required in generating recommendations for G, GroupReM applies a *blocking strategy*[3] on movies archived at W to obtain the subset of movies that are potentially of interest to the members of G (to various degrees), denoted *Candidate_Movies*, to be considered for recommendation.

The blocking strategy adopted by GroupReM first considers the personal tags assigned by a group member U of G for each of his/her bookmarked movies, uM. A movie M archived at W [4] is included in *Candidate_Movies* if *each* of the personal tags assigned by U to uM exactly matches or is highly similar to at least a tag in the tag cloud of M. As tags are concise and valid content descriptors of an item [GZR+10], it is anticipated that movies in *Candidate_Movies* are of interest to (at least one of the members of) G, since each movie in *Candidate_Movies* shares a number of same (or analogous) tags (to a certain degree) with the ones in the group profile for G.

To select movies to be included in *Candidate_Movies*, GroupReM relies on a reduced version of the word-correlation matrix (introduced in Section 3.2) which contains 13% of the most frequently-occurred words (based on their frequencies of occurrence in the Wikipedia documents), and for the remaining 87% of the less-frequently-occurring words, only the exact-matched correlation factor, i.e., 1.0, is used [GN08]. By adopting a reduced version of the word-correlation matrix to determine potentially similar movies, the overall processing time of GroupReM is significantly reduced without affecting its accuracy [PLN09].

Example 2 Consider the five movies archived at MovieLens, i.e., ML_1, ML_2, ML_3, ML_4, and ML_5, as shown in Figure 3, which are not bookmarked by any member of the group shown in Figure 2. To determine which one of the five movies should be treated as candidate movies for the group G introduced in Example 1, GroupReM compares personal tags assigned to each movie shown in Figure 2 with the tags (in the tag cloud) of each movie shown in Figure 3. Given that each of the personal tags assigned to M_1 is highly similar to at least a tag in the tag cloud of ML_1, i.e., the word-correlation factors of "family" and "Disney" ("animation" and "cartoon", respectively) can be found in the reduced version of the word-correlation matrix, ML_1 is selected as a candidate movie. Furthermore, each of the personal tags assigned to describe M_3 (M_6, respectively) exactly matches its counterpart in ML_4 (ML_3, respectively). Therefore, ML_4 (ML_3, respectively) is a candidate movie. In addition, since the "drama" tag in M_7 exactly matches its counterpart in (the tag cloud of) ML_5 and the remaining personal tag of M_7, i.e., "family", is highly similar to

[3] A blocking strategy is a filtering technique that reduces the potentially very large number of comparisons to be made among records [Chr08], i.e., movies available at a movie website in our case.

[4] A movie archived at a movie website is included in the set of candidate movies if it has not been bookmarked by any member of a given group.

ML_1: Disney (16), cartoon (2), lion (1), ...
ML_2: drama (2), cancer (1), youth (1), ...
ML_3: drama (3), Vietnam (9), war (5), ...
ML_4: courtroom (3), classic (1), drama (4), ...
ML_5: poverty (2), oscar (8), drama (2), ...

Figure 3: (Portions of the) Tag clouds of potential candidate movies in which tags exactly-matched or highly similar to the personal tags assigned to a movie shown in Figure 2 are underlined

another tag in (the tag cloud of) ML_5, i.e., "poverty", ML_5 is also selected as a candidate movie. Although M_2, M_3, M_6, and M_7 include a tag, i.e., "drama", which is also a tag in the tag cloud of ML_2, none of the remaining personal tags assigned to describe the content of either M_2, M_3, M_6, or M_7 exactly matches or is similar to a tag in ML_2, and thus ML_2 is not treated as a candidate movie. $\Box$

3.4 Generate Group Recommendations

Having identified the set of candidate movies to be considered for recommendation to a group, GroupReM proceeds to rank each of the candidate movies by relying on two different scores, the *group appealing* and *popularity* scores, presented in Sections 3.4.1 and 3.4.2, respectively. The two scores are combined using an *aggregation function*, as defined in Section 3.4.3, and the top-10 candidate movies with the highest combined scores are recommended to the group.

3.4.1 Appealing Scores of Movies

To determine the degrees of interests of members in a group G on a candidate movie M, GroupReM computes the *group appealing* score of M for G, denoted *GrpApp*(M, G), by accumulating the word correlation factors among the tags that capture the types of movies members of G are interested in, i.e., tags in the group profile of G, and tags in the tag cloud of M. In computing the *GrpApp* score of M for G, GroupReM relies on the word-correlation matrix introduced in Section 3.2, instead of the reduced word-correlation matrix employed in Section 3.3, since the former provides a more accurate similarity measure between (tags representing) M and G than the reduced matrix. The *GrpApp* score of M for G is defined as

$$GrpApp(M, G) = \sum_{g \in GP} \sum_{m \in M} wcf(g, m) \times \frac{freq_g}{Max(freq_{GP})} \times \frac{freq_m}{Max(freq_M)} \qquad (1)$$

where g (m, respectively) is a tag in the group profile GP of G (the tag cloud of M, respectively), $wcf(g, m)$ is the word-correlation factor of g and m in the word-correlation matrix , $freq_g$ ($freq_m$, respectively) is the frequency of occurrence of tag g (m, respectively) in GP (the tag cloud of M, respectively), $Max(freq_{GP})$ ($Max(freq_M)$, respectively) is the highest frequency of any tag in GP (the tag cloud of M, respectively), and $\frac{freq_g}{Max(freq_{GP})}$ ($\frac{freq_m}{Max(freq_M)}$, respectively) denotes the *weight* of g (m, respectively).

$Freq_g$ ($freq_m$, respectively) in Equation 1 is an indicator of the relative *degree of significance* of tag g (m, respectively) in representing (the content of) GP (M, respectively), since it reflects the frequency in which group members (number of users at a movie website, respectively) have chosen g (m, respectively) to represent movies of interest for G (the content of M, respectively). The *larger* $freq_g$ ($freq_m$, respectively) is, the more *significant* g (m, respectively) is in characterizing GP (describing M, respectively). In addition, by relying on the *weight* of each tag in GP (the tag cloud of M, respectively) in Equation 1, GroupReM ensures that exactly-matched (or highly-similar) tags between GP and M do not inflate the group appealing score of M if they are not significant/representative tags to G (M, respectively).

Example 3 To illustrate the merit of using word-correlation factors in computing the *GrpApp* score of a candidate movie, consider the group profile shown in Figure 2 and the candidate movies ML_1 and ML_5 as shown in Figure 3. Both movies include a tag, i.e., "Disney" and "drama", respectively in their corresponding tag clouds that exactly matches its counterpart in the group profile of G (as shown in Figure 2), which implies that the *GrpApp* score of ML_1 and ML_5 should be similar. Taking into account the remaining, i.e., non-exact-matched but analogous, tags in the tag clouds of the aforementioned movies in calculating their respective *GrpApp* scores, GroupReM computes a more accurate group appealing score for each candidate movie. *GrpApp*(ML_1, G), computed using Equation 1, is 3.5, whereas *GrpApp*(ML_5, G) is 1.0, which correctly reflects that G, as a whole, is more interested in *family*, *animated* movies than *dramatic* movies, as captured in the group profile of G. $\Box$

3.4.2 Popularity Scores of Movies

In addition to computing the *GrpApp* score of a candidate movie M for G, GroupReM also considers the *global popularity* score of M, denoted *GlbPop*(M), which exploits the "wisdom of the crowd" [BFC09], i.e., the collective interest in M expressed by users at the movie website of which members of G are users, and provides a higher ranking on M if it is *more frequently* bookmarked at the website than other candidate movies.

Popular movies which attract the attention of users at a movie website are more likely to be bookmarked by the users. GroupReM weights the fact that frequently-bookmarked movies may also be of interest to members of G. While solely relying on the popularity of an item in performing the recommendations task (which does not apply to GroupReM) can lead to less diverse and useless recommendations [ZWZ+10], Adomavicius and Kwon [AK11] claim that the accuracy of the recommendations can be enhanced by considering the popularity of an item during the recommendation process.

GlbPop, which is considered by GroupReM as an additional decision factor besides *GrpApp* to rank M to make recommendations, is computed as the *total number* of users at W who have *bookmarked* M.

3.4.3 Rank Aggregation

Having determined the group appealing and global popularity scores of each movie M in *Candidate_Movies*, GroupReM computes the *ranking score* of M by applying a popular linear combination measure, called $CombMNZ$ [Lee97], which is frequently used in fusion experiments [CCB09]. CombMNZ considers multiple existing lists of rankings on an item I to determine a joint ranking of I, a task known as rank aggregation or data fusion.

$$CombMNZ_I = \sum_{c=1}^{N} I^c \times |I^c > 0| \tag{2}$$

where N is the number of ranked lists to be fused, i.e., the number of input ranked lists, I^c is the normalized score of I in the ranked list c, and $|I^c > 0|$ is the number of non-zero, normalized scores of I in the lists to be fused.

Prior to computing the ranking score of M, it is necessary to transform the original scores in each individual ranked list into a *common range*, which can be accomplished by applying Equation 3 to each score in each ranked list so that it is within the range [0, 1], a common range [Lee97].

$$I^c = \frac{S^I - I^c_{min}}{I^c_{max} - I^c_{min}} \tag{3}$$

where S^I is the score of item I in the ranked list c prior to be normalized, I^c_{max} (I^c_{min}, respectively) is the maximum (minimum, respectively) score available in c, and I^c is the normalized score for I in c.

GroupReM normalizes the group appealing and global popularity scores of M computed in Sections 3.4.1 and 3.4.2, respectively using Equation 3. Thereafter, using CombMNZ, GroupReM (i) sets $N = 2$ (in Equation 2), which is the number of (input) ranked lists of normalized scores with the original ones computed in Sections 3.4.1 and 3.4.2, respectively, (ii) determines the *overall ranking* score of each movie M in *Candidate_Movies* using Equation 2, and (iii) recommends the top-10 ranked movies to (the members of) G.

By adopting this fusion strategy, GroupReM considers the strength of each evidence, i.e., the *GrpApp* and *GlbPop* scores, as opposed to simply positioning higher in the ranking movies with a high *GrpApp* or *GlbPop* score.

Example 4 Consider the candidate movies ML_1, ML_3, ML_4, and ML_5 as shown in Figure 3, along with their respective (normalized) group appealing and global popularity scores as shown in Table 1. Using CombMNZ as a rank aggregation measure, GroupReM identifies the most relevant movies, i.e., movies of interest, for G. Even though the (normalized) global popularity score of ML_1 is slightly lower than the global popularity score of ML_3, GroupReM positions ML_1 higher than ML_3 in the ranking of movies to be recommended. This is because ML_1 is more appealing for (members of) G based on the tags in the tag cloud of ML_1 and the tags in the group profile of G that depict the movie preferences of (the members of) G.

As shown in Table 1, the global popularity score of ML_5 is relatively high; however, its group appealing score is significantly lower in comparison with the group appealing scores of the remaining candidate movies. As a result, GroupReM positions ML_5 lower in the ranking of movies to be recommended than the remaining candidate movies in Figure 3. □

Candidate Movie	Group Appealing Score	Global Popularity Score	Ranking
ML_1	0.92	0.65	**3.14**
ML_3	0.71	0.80	3.02
ML_4	0.53	0.56	2.18
ML_5	0.27	0.75	2.04

Table 1: Normalized scores for the candidate movies shown in Figure 3 with respect to the group profile of G shown in Figure 2 as computed by GroupReM

# of Distinct Users	2,113
# of Distinct Movies	10,197
# of Distinct Tags	13,222
# of Distinct Tag-Movie Assignments	47,957
Average # of Movies Bookmarked per User	13
Average # of Distinct Tags Assigned to Movies per User	23
Average # of Distinct Tags Assigned to a Movie	8
Average # of Ratings Assigned to Movies per User	405
Average # of Ratings Assigned to a Movie	85

Table 2: Statistical information of the MovieLens dataset

4 Experimental Results

In this section, we first introduce the dataset (in Section 4.1) employed for assessing the performance of GroupReM. Thereafter, we present the evaluation protocol and group formation strategy adopted for creating the groups used for the evaluation purpose (in Sections 4.2 and 4.3, respectively). We define the metric which quantifies the accuracy and ranking approach of GroupReM (in Section 4.4). We detail the empirical study conducted for verifying the effectiveness and efficiency of GroupReM and compare its performance with existing group recommenders on movies (in Section 4.5).

4.1 Dataset

To evaluate GroupReM in recommending movies appealing (to a certain degree) to the members of a group, we consider the MovieLens dataset [aUoM], a dataset released by the ACM HetRec Conference in 2011. Statistical information on MovieLens is shown in Table 2. (See detailed information on the dataset at grouplens.org/system/files/hetrec2011-movielens-readme.txt.) Note that the MovieLens dataset was not developed for assessing the performance of group recommenders, since pre-defined groups of users are not provided in the dataset. For this reason, we create our own groups of users for the evaluation purpose (see details in Section 4.3).

4.2 Evaluation Protocol

To assess the relevancy of group recommendations suggested by GroupReM, we have adapted a standard approach to partition the movies bookmarked by each user in the MovieLens dataset into two subsets and employed the five-fold cross validation approach [MS03]. In evaluating the recommendations made by GroupReM for a given group G, in each of the five repetitions, 80% of the movies bookmarked in MovieLens by each member U of G were treated by GroupReM as included in the individual profile of U and the remaining 20% were reserved for the testing purpose, i.e., to assess the relevance of the recommendations generated for (U in) G. A recommendation made by GroupReM is treated as *relevant* for (U in) G, if the recommended movie is included in the 20% of the movies (bookmarked by U) withheld for the testing purpose, a commonly-employed protocol for assessing recommendation systems [BCC10, GWB+10].

4.3 Group Formation

To the best of our knowledge, there are no benchmark datasets available for assessing the performance of group recommenders, needless to say group recommenders on movies. For this reason, we employ a popular strategy for generating groups (of users in the MovieLens dataset introduced in Section 4.1) for evaluation purpose.

In creating groups for evaluating the recommendations generated by GroupReM, we consider two important factors: the *size* and *cohesiveness* of a group [AYRC+09, BMR10]. By varying the *group sizes*, we can assess the difficulty in reaching *consensus* among members of small versus large groups. We consider groups with 2 to 8 members, which are comparable to the group sizes defined in [AYRC+09, BMR10], to demonstrate the effectiveness of GroupReM in recommending movies for small, as well as large, groups.

Besides group size, *group cohesiveness* is another important criterion [AYRC+09] in evaluating group recommenders. By using groups that include members with various degrees of cohesiveness, i.e., different degrees of user-to-user similarity, we can verify the correctness of GroupReM in generating recommendations for groups of users that may or may not share common preferences in movies, since the latter is more challenging than the former in terms of satisfying their mutual interest. Altogether, three different types of groups, i.e., *highly similar*, *dissimilar*, and *random*, are considered. *Random groups* are formed by randomly selecting users from MovieLens, regardless of their preferences on movies. *Highly-similar* groups include members with common interests in the same types of movies, whereas *dissimilar groups* reflect groups of people that are different in terms of their preferences in movies. To determine the users who should be included in highly-similar and dissimilar groups, we adapted the strategy employed in [BMR10], which calculates the user-to-user similarity, denoted *User_Sim*, on each pair of users in MovieLens. The *User_Sim* metric is introduced in [AYRC+09] and computed as

$$User\_Sim(u, u') = \frac{|\{i \mid i \in I_u \bigwedge i \in I_{u'} \bigwedge |rating(u, i) - rating(u', i)| \geq 2\}|}{|\{i \mid i \in I_u \bigvee i \in I_{u'}\}|} \quad (4)$$

where I_u ($I_{u'}$, respectively) denotes the set of items, i.e., movies in our case, rated by user u (u', respectively), i is an item, $rating(u, i)$ ($rating(u', i)$, respectively) denotes the rating

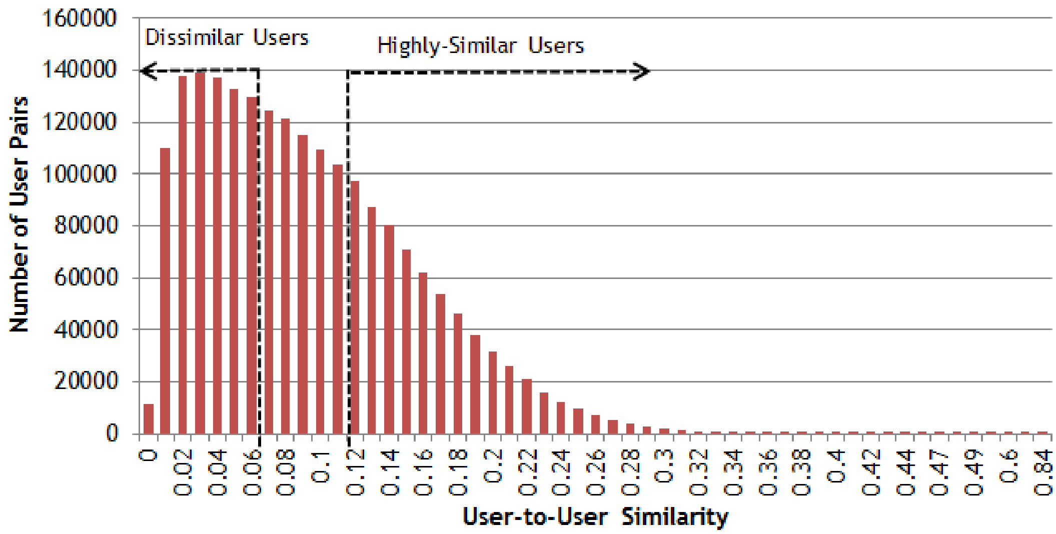


Figure 4: User-to-user similarity distribution in the MovieLens dataset

assigned to i by u (u', respectively), and $|rating(u,i) - rating(u',i)| \geq 2$ constraints a movie M to be treated as "shared" between u and u' if they both rated M within 2 units of each other on the scale of 0 to 5, which indicates that u and u' assigned a similarly high (low, respectively) rating to M. The high (low, respectively) similar ratings provided by u and u' on i indicates that u and u' share the same preference on i.

In computing the $User\_Sim$ score between any two users (as defined in Equation 4), only pairs of users who have rated at least 5 common items are considered, a common practice among CF-based recommenders which ensures that the correlation between two users, i.e., $User\_Sim$ score, is not high (low, respectively) solely based on the same ratings assigned to a small set of items, i.e., less than 5 movies in our case, by the two users [BMR10].

We follow the strategy proposed by the authors in [BMR10], who consider the distribution of user pairs in a given dataset (based on their user-to-user similarity) and treat the 33% of user-pairs with the highest user-to-user similarity score as *highly-similar* users. Based on the distribution of user-to-user similarity scores for each pair of users in MovieLens (as shown in Figure 4), we observe that pairs of users with a 0.11 $User\_Sim$ score or higher fall within the range of 33% user-pairs who achieve the highest user-to-user similarity. Hence, a group of MovieLens users whose user-to-user similarity among each other is higher or equal to 0.11 is treated as a *highly-similar* group. Applying the same strategy to determine highly-similar users, we treat the 33% of users-pairs with the lowest $User\_Sim$ scores as dissimilar users. As it turns out, user-pairs with a $User\_Sim$ score less than or equal to 0.06 constitute the 33% of user-pairs in MovieLens with the lowest user-to-user similarity (as shown in Figure 4), and these users are treated as members of *dissimilar* groups.

Based on the group formation protocol defined above, we created 3,150 distinct groups, which are uniformly distributed among highly-similar, dissimilar, and random groups. In addition, each set of the 1,050 groups that share the same degree of cohesiveness is uniformly distributed based on the pre-defined group sizes, i.e., 2 to 8 members. Thus, for each distinct group size there are 150 groups in which group members share the same (pre-determined) degree of cohesiveness.

4.4 Metrics

To assess the overall performance and ranking strategy of GroupReM, we employ the Normalized Discounted Cumulative Gain (*nDCG*) [CMS10] measure, which is a standard IR metric often used for evaluating group recommenders [AYRC+09, BMR10]. Due to the lack of "ground truth" required to assess the recommendations generated by GroupReM for a given group G of a particular size that includes members (without) sharing the same degree of cohesiveness, we calculate the $nDCG$ for G as the *average* of the $nDCG$ value computed for each of the group members in G, following the experimental setting adopted by Amer-Yahia et al. [AYRC+09].

$nDCG_{10}$, as defined in Equation 5 for evaluating the relevance of each batch of *top-10* recommendations generated by GroupReM, *penalizes* relevant movies ranked *lower*. The penalization is based on a relevance reduction, which is logarithmically proportional to the relative position of each relevant movie in a ranked list of recommended movies (as shown in Equation 6). The *higher* the $nDCG_{10}$ score is, the *better* the ranking strategy adopted by the corresponding recommender system RS is, since a high $nDCG_{10}$ score on a list of recommendations L indicates that relevant recommendations generated by RS are positioned high in L.

$$nDCG_{10} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{k=1}^{M}\frac{DCG_{10,k}}{IDCG_{10,k}} \tag{5}$$

where N (which is 150 in our case) is the number of groups with a pre-defined number of group members such that the members share the same pre-determined degree of cohesiveness (as detailed in Section 4.3), i is the i^{th} group for which GroupReM generates movie recommendations, M is the number of group members in i, k is the k^{th} group member in i, $IDCG_{10,k}$ (in Equation 5) is the best possible $DCG_{10,k}$ value for the recommendations generated by GroupReM for k[5], and

$$DCG_{10,k} = \sum_{j=1}^{10}\frac{(2^{rel_j}-1)}{log_2(1+j)} \tag{6}$$

where rel_j is the binary relevant judgment of the recommended movie at the j^{th} ranking position and is assigned a value of "1" if the movie is a *relevant* recommendation for k (as defined in Section 4.2) and is assigned a "0", otherwise.

4.5 The Effectiveness and Efficiency of GroupReM

In this section, we first verify the correctness of relying on *word-correlation factors* and the *popularity* of movies to generate group recommendations (as presented in Section 4.5.1).

[5] $IDCG_{10,k}$ is computed as $DCG_{10,k}$ using an *ideal* ranking such that the ten recommendations are arranged in descending order based on their relevant judgment scores in the ranked list.

Thereafter, we compare the *performance* of GroupReM with existing CF-based group recommenders (in Section 4.5.2) and assess the *efficiency* of GroupReM and CF-based group recommenders in performing the recommendation task (in Section 4.5.4).

4.5.1 The Correctness of GroupReM

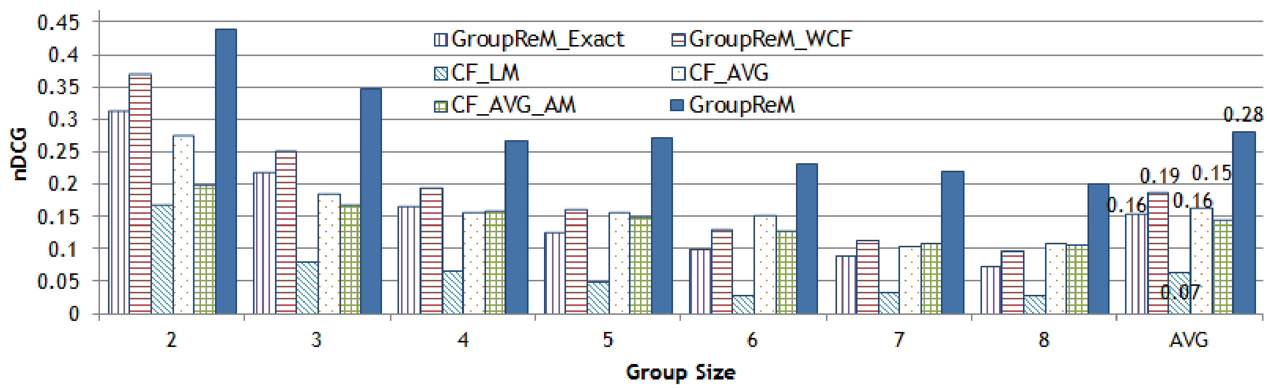
As stated in Section 3.4, GroupReM depends on the *group appealing* (based on word-correlation factors) and *global popularity* scores to generate recommendations of interest to a group. To verify the effectiveness of GroupReM in making group recommendations on movies, we conducted an empirical study in which we compared two alternative implementations of GroupReM. The first alternative, denoted *GroupReM_Exact*, relies solely on the group appealing score computed on exactly-matched tags for generating movie recommendations for a group G. In this case, the group appealing score of a candidate movie M for G is calculated using the *Dice* coefficient [CMS10] on the tags in (the tag cloud of) M and the tags characterizing (the group profile of) G. The second alternative, denoted *GroupReM_WCF*, relies on the word-correlation factors and considers analogous, besides exactly-matching, tags. GroupReM_WCF computes the group appealing score of each candidate movie using Equation 1.

As illustrated in Figure 5, regardless of the degree of cohesiveness among group members in groups of any size, GroupReM_WCF consistently improves the accuracy of the recommendations generated by GroupReM_Exact. The 3% overall improvement on the (average) $nDCG$ achieved by GroupReM_WCF over GroupReM_Exact, using the MovieLens dataset and the groups introduced in Section 4.3, indicates that relaxing the exact-matching constraint by adopting word-correlation factors enhances the accuracy of movies recommended to a group by GroupReM_WCF. In addition, at least 8% overall improvement on the (average) $nDCG$ scores achieved by GroupReM over GroupReM_WCF, using the aforementioned dataset, validates the fact that the global popularity score (as defined in Section 3.4.2) further increases the accuracy of group recommendations than simply using the group appealing scores of movies to perform the group recommendation task (as illustrated in Figures 5(a)-5(c)). Note that the differences between GroupReM_WCF and GroupReM_Exact with respect to GroupReM, in terms of $nDCG$, are statistically significant, as determined using a Wilcoxon Rank Sum Test ($p < 0.05$).

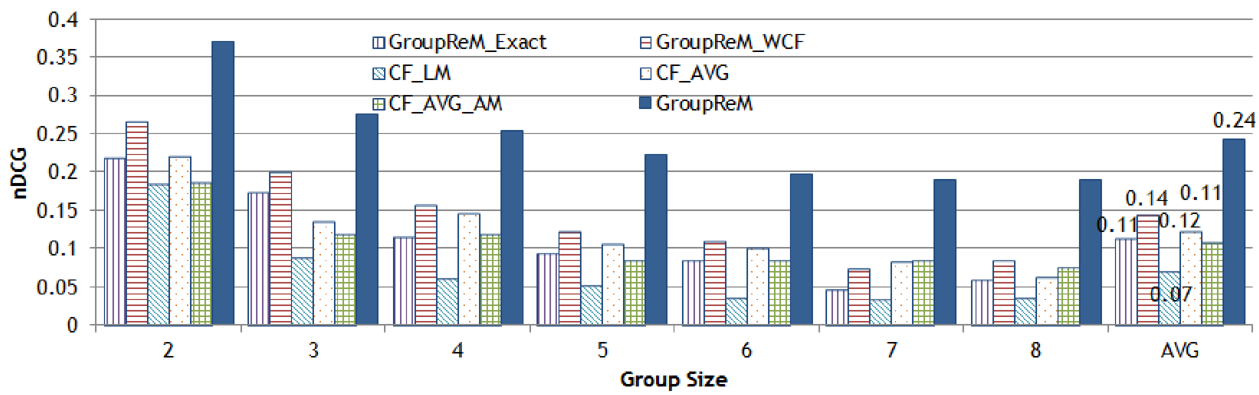
4.5.2 Comparing the Performance of GroupReM with Existing Group Recommenders

To further verify and demonstrate the *effectiveness* of GroupReM, we compare its performance with two well-known CF recommenders on movies, which are based on *Average* (CF_AVG) and *Least Misery* (CF_LM) aggregation strategies [AYRC+09, BMR10], respectively. Given that GroupReM adopts an aggregated model approach to make recommendations, we also compare its performance with a CF recommender that employs an *average* aggregated model strategy (CF_AVG_AM). We have chosen CF-based recommenders for comparisons, since to the best of our knowledge there is no group recommender *on movies* that depends primarily on content descriptions to make recommendations.

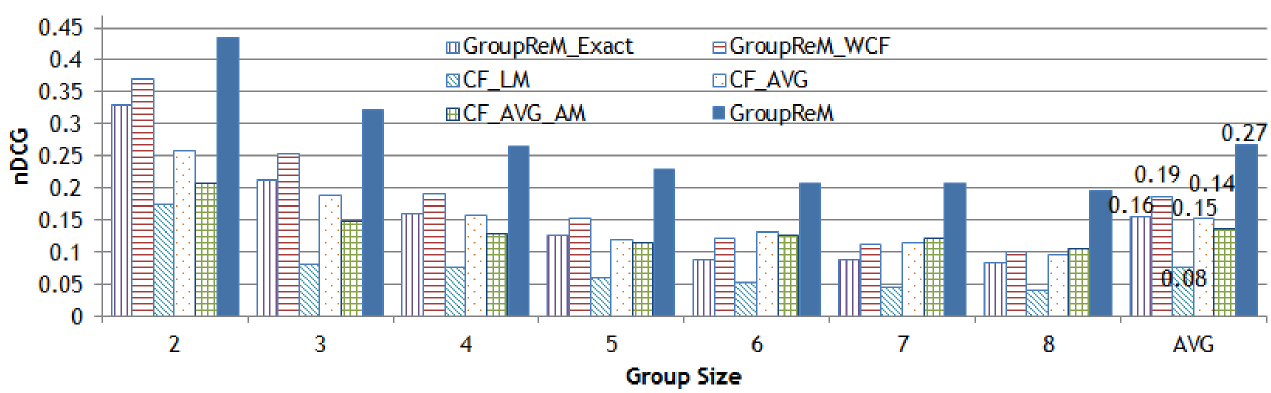
Given a group G, both CF_AVG and CF_LM first generate movie recommendations for individual members of G by employing the well-known CF strategy. Thereafter, the recom-



(a) Groups of *highly-similar* users



(b) Groups of *dissimilar* users



(c) Groups of *random* users

Figure 5: (Average) *nDCG* scores computed for (the alternative implementations of) GroupReM and alternative implementations of the collaborative filtering approach based on *average* and *least misery aggregation* strategies on 3,150 groups of various sizes. All the differences in *nDCG* are statistically significant with respect to GroupReM (Wilcoxon, $p < 0.05$).

menders proceed to merge the recommendations generated for individual group members to create the list of movies to be recommended to G. While CF_AVG computes the score of a movie M for G by *averaging* the ratings of M predicted for each individual group member in G, CF_LM defines the score of M for G as the smallest predicted rating of M among all the rating predictions of M determined for each of the individual members of G. The top-10 movies with the highest ratings are recommended to G. (A more in-depth discussion on CF_AVG and CF_LM can be found in [AYRC$^+$09, BMR10].) The CF_AVG_AM approach, on the other hand, generates a single group profile by averaging the ratings of each movie bookmarked by each individual member of G. Thereafter, the well-known CF approach is employed to generate a list of the top-10 highest ranked movies for (the profile of) G.

Prior to comparing the performance of the aforementioned recommenders with GroupReM, we have determined the relevance of each movie recommended by CF_AVG, CF_LM, and CF_AVG_AM for each of the groups constructed in Section 4.3 using the MovieLens dataset, evaluation protocol, and metric detailed in Sections 4.1, 4.2, and 4.4, respectively.

Figures 5(a), 5(b), and 5(c) show the $nDCG$ scores achieved by GroupReM, CF_LM, CF_AVG, and CF_AVG_AM for highly-similar, dissimilar, and random groups of different sizes, respectively. The average $nDCG$ score of GroupReM computed for groups with *highly-similar* users is 0.28, which is at least 12% higher than the average $nDCG$ scores achieved by either CF_LM, CF_AVG, or CF_AVG_AM, which are 0.07, 0.16, and 0.14, respectively. The average $nDCG$ score achieved by GroupReM for groups with *dissimilar* (*random*, respectively) users is 0.24 (0.27, respectively), which also outperforms the average $nDCG$ scores achieved by CF_LM, CF_AVG, and CF_AVG_AM on the same groups, which are 0.07, 0.12, and 0.11 (0.08, 0.15, and 0.14, respectively). All of these $nDCG$ values achieved by GroupReM are statistically significant over CF_LM, CF_AVG, and CF_AVG_AM (as verified using a Wilcoxon Rank Sum Test for $p < 0.05$).

A higher $nDCG$ value indicates that GroupReM is more effective than CF_LM, CF_AVG, and CF_AVG_AM in detecting and ranking higher in the list of recommended movies the ones that are relevant, i.e., of interest, to a group, regardless of the number of members in the group or the similarity among group members in terms of their preferences in movies.

4.5.3 Observations

Since only movies reserved for the testing purpose (as detailed in Section 4.2) are considered relevant, it is not possible to account for the potentially relevant movies that the users have not bookmarked. As a result, the $nDCG$ scores in our empirical study are underestimated, which is a well-known limitation of the evaluation protocol (introduced in Section 4.2) applied to recommender systems [HKBR99]. As this limitation affects all the evaluated recommenders, i.e., (alternative implementations of) GroupReM, CF_AVG, CF_LM, and CF_AVG_AM, the $nDCG$ values are consistent for the comparative evaluations [BCC10].

Regardless of the degrees of cohesiveness among group members, the $nDCG$ scores computed for GroupReM (CF_AVG, CF_LM, and CF_AVG_AM, respectively) consistently *decrease* when the group *size increases*. This decrease in $nDCG$ score is expected as *more* users are involved in a group, the *harder* it is to reach consensus among members in terms of choosing movies that represent the collective interests of the group. Moreover, regardless of the size of the groups under evaluation, the *nDCG* scores computed for GroupReM (CF_AVG, CF_LM, and CF_AVG_AM, respectively) are slightly *higher* when considering groups with *highly-similar* users. This is anticipated, since the *more similar* the group members are with one other in terms of their preferences in movies, the *more likely* they will treat each recommendation the same, i.e., as (non-)relevant. The results of the analysis on the performance of GroupReM (and other recommenders used for comparison purposes), in terms of the degree of cohesiveness among group members, correlates with the empirical study conducted in [AYRC$^+$09, BMR10].

Note that the fact that CF_AVG_AM and CF_AVG outperform CF_LM is anticipated, since the latter adopts a least misery strategy which favors the "least happy" group member in making recommendations. Furthermore, CF_AVG, CF_LM, and CF_AVG_AM rely on identifying "similar-minded" users within a movie community, i.e., a movie website, to generate movie recommendations. The search is applied to each member of a given group G. In doing so, CF_AVG, CF_LM, and CF_AVG_AM solely consider users of a movie website who rate the same movies as the ones that have been bookmarked and rated by members of G. Hence, the *less* "similar-minded" the users are (with respect to a member U of G), the *less* reliable are the ratings predicted for movies to be recommended to U (and G). GroupReM, on the other hand, does *not* require locating "similar-minded" users to perform the recommendation task. Instead, GroupReM, relies on content-similarity on tags and the popularity scores of the candidate movies.

4.5.4 Efficiency of GroupReM

Besides assessing the effectiveness of GroupReM, CF_AVG, CF_LM, and CF_AVG_AM on making movie recommendations to a group (in Section 4.5.2), we have also validated the overall *efficiency* of (the variations of) GroupReM, CF_AVG, CF_LM, and CF_AVG_AM in suggesting movies of interest to a group.

Figure 6 shows the average time (in seconds) required for (the alternative implementations of) GroupReM, CF_AVG, CF_LM, and CF_AVG_AM to generate recommendations for the 1,050 groups of various sizes, such that group members share the same degree of cohesiveness among one another, using the 5-fold evaluation strategy detailed in Section 4.2. While GroupReM_Exact achieves the shortest processing time, which is 68 seconds, the additional processing time required by GroupReM, which is 66 (= 134-68) seconds, is relatively insignificant, compared with the degree of accuracy achieved by GroupReM in generating recommendations of interest to a group, as shown in Section 4.5.1.

GroupReM and CF_AVG_AM require similar processing time to generate recommendations. When compared with CF_AVG and CF_LM, however, GroupReM requires significantly less time, i.e., as illustrated in Figure 6, the processing time of CF_AVG and CF_LM increases by at least *8 minutes* in comparison with the processing time of GroupReM.

To further assess the efficiency of GroupReM, we consider 450 (= 3 × 15) groups (regardless of the degree of cohesiveness among the members of the group) of MovieLens users of each pre-defined size, i.e., 2 to 8, for evaluation purpose (as detailed in Section 4.3). We computed the average processing time of GroupReM in generating recommendations for each one of the 450 groups of pre-defined size. As illustrated in Figure 7, the (average) time (in milliseconds) required by GroupReM to generate group recommendations does not ex-

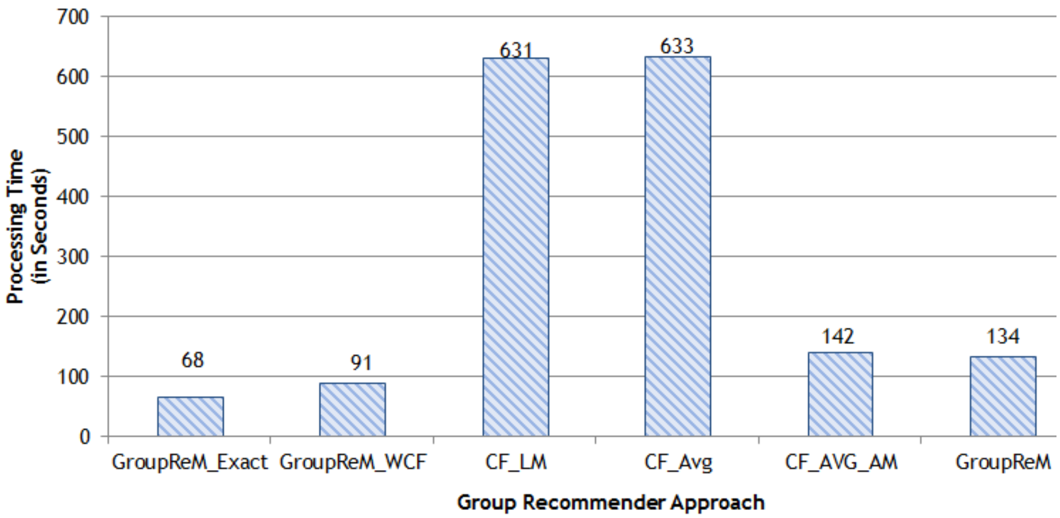


Figure 6: (Average) Time for (alternative implementations of) GroupReM, CF_AVG, CF_LM, and CF_AVG_AM to generate recommendations for 1,050 groups (regardless of their sizes), such that group members share the same degree of cohesiveness among each other, using the 5-fold evaluation strategy detailed in Section 4.2

ponentially increase when the number of group members increases. Instead, as determined by the curve created using the Microsoft Excel Trend/Regression tool (also shown in Figure 7), the increase in processing time of GroupReM when the number of group members increases follows a *linear* trend, which demonstrates the scalability of GroupReM.

We have also evaluated whether the total number of movies bookmarked by the members of a group can significantly affect the group recommendation processing time of GroupReM. To draw a conclusion, we considered the 3,150 groups defined Section 4.3 and calculated the processing time of GroupReM in generating recommendations for each of the groups, regardless of the size of the groups or the degree of cohesiveness among group members. As anticipated, the processing time (in milliseconds) required for GroupReM to generate recommendations *increases* as the total number of movies bookmarked by group members *increases*, as illustrated in Figure 8. However, even though the total number of movies bookmarked by group members is in the thousands, the processing time of GroupReM in suggesting movies of interest to a group is at most 2.5 seconds, which is a relatively short period of time. Furthermore, the increase in processing time follows a *polynomial* trend, as determined by the curve created using Microsoft Excel Trend/Regression tool and as shown in Figure 8.

Note that independently of the 3,150 groups introduced in Section 4.3, we have empirically evaluated GroupReM on generating recommendations for groups of up till 100 members. Based on the conducted experiments, we have observed that (i) the total number of movies bookmarked by group members remains in the thousands and (ii) the processing time of GroupReM is at most 5 seconds, even when considering groups of approximately 100 members with thousands of movies bookmarked among them.

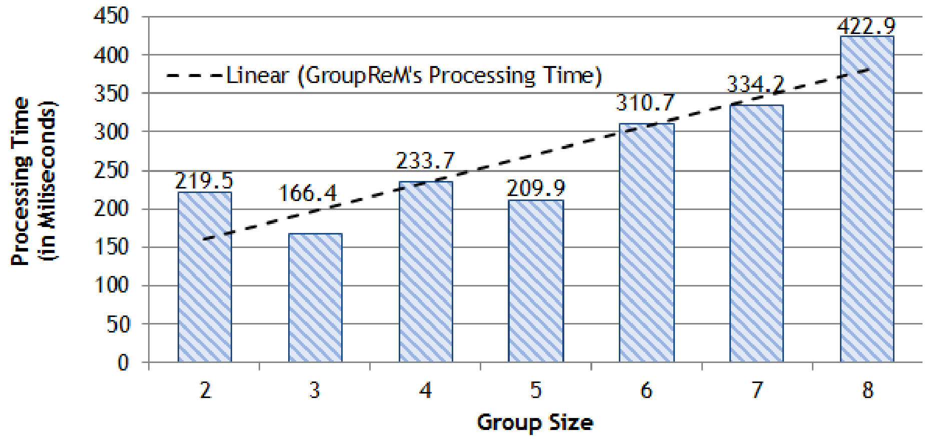


Figure 7: (Average) Processing time of GroupReM for generating movie recommendations for groups including a certain number of group members, which is computed using groups of a pre-defined size, i.e., 2 to 8, as defined in Section 4.3

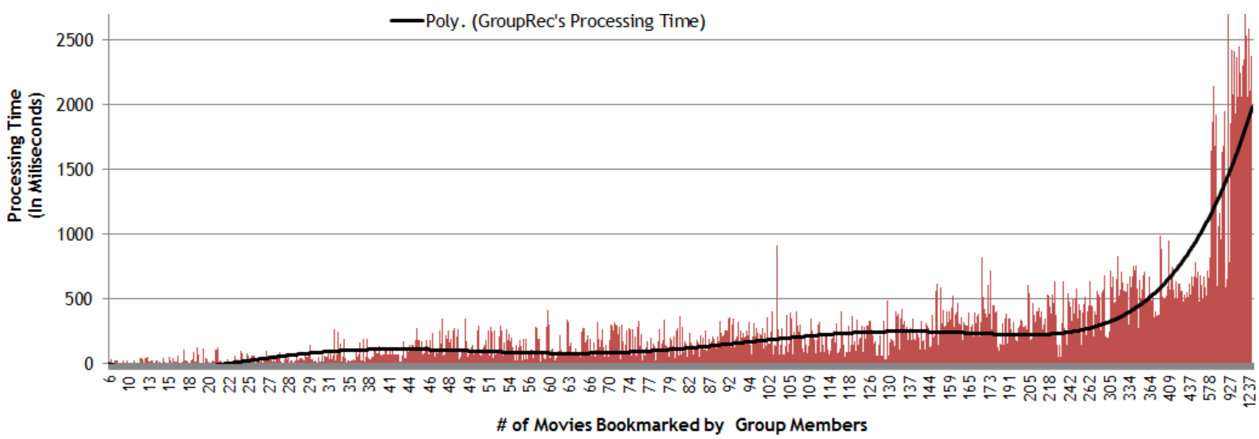


Figure 8: Average time for GroupReM to generate group recommendations for groups with different number of bookmarked movies among group members, which is computed using 3,150 groups created in Section 4.3

4.6 Limitations of the Current Implementation of GroupReM

GroupReM, as currently developed, adopts a Top-N strategy and suggests a list of N movies to a group of users at a given time [DK04]. The current design of GroupReM does not consider the dynamic preferences of group members that may evolve over time. Moreover, the satisfaction of a group member U on the recommended items, i.e., movies in our case, may depend on other group members. As stated in [BMR10, MG06], U can be influenced by other group members through emotional contagion and conformity. The former claims that U's satisfaction may be increased if other group members are satisfied with the recommendations, whereas the latter states that the opinions of other users may influence U's opinions. The recommendation strategy adopted by GroupReM, however, does not consider that some members of a group are more capable than others to influence the remaining group members in making decisions on the (non-)relevance of movies suggested to the group, an issue to be addressed as future work.

5 Conclusions and Future Work

With the popularity of social activities in which groups of people are involved, either online or in person, group recommenders that are designed for identifying items of interest to a group play a significant role in social networking. One of the item domains that predominates on group recommenders is movies. Groups of friends, family members, and acquaintances, who gather to watch a movie at home or at the cinema, can use the service of a group recommender to find movies pertaining to their interests. Identifying movies to be recommended that appeal a group, however, is a non-trivial task due to the personal (and often diverse) preferences of group members in movies. We have introduced GroupReM, a group recommender on movies, which advances the current technology in solving the problem.

To suggest movies for members of a given group G at a movie website W, GroupReM first constructs a *group profile* for G, which captures the collective interests of members of G in movies. Hereafter, GroupReM relies on a simple *aggregation model* to determine the ranking score of each candidate movie M archived at W, which has not been bookmarked by members of G and is potentially of interest to G, based on the (i) *content similarity* between M and the group profile of G and (ii) *popularity* of M at W so that the top-10 ranked movies are recommended to G.

Unlike existing group recommenders on movies, which are based on the collaborative-filtering (CF) strategy and rely solely on the ratings assigned to movies to perform the recommendation task, GroupReM takes the advantage of the richness of semantic information, i.e., (personal) tags, which are available at any movie website. Considering the content-similarity of movies and a group profile, GroupReM is not constrained to find users at a movie website who are "similar-minded" based on ratings assigned to the same movies to suggest movies to a group, as CF-based group recommenders do. In addition, GroupReM employs word-correlation factors and considers non-exact-matched, but analogous, tags to more adequately determine the degree of appeal of a movie to a group, which in turn enhances the accuracy of the recommendations.

We have conducted an empirical study using more than 3,000 groups of various sizes and degrees of cohesiveness among group members, who are users in the MovieLens dataset, to verify the *effectiveness* and *efficiency* of GroupReM. The experimental results indicate that GroupReM is highly accurate in suggesting movies appealing (to a certain degree) to the members of a group. We have compared the performance of GroupReM with three well-known CF-based recommenders and verified that GroupReM outperforms the aforementioned recommenders by a large margin, and the *average* processing time of GroupReM is significantly shortened in comparison to its counterparts.

GroupReM relies on personal tags assigned to movies that have been bookmarked by group members to create group profiles and identify movies to be recommended. Occasionally, personal tags may not be available or they may be too broad in describing (the content of) a movie. We plan to investigate strategies that can be applied to *infer* tags that adequately represent the content of movies, if personal tags are missing or too general, which can further enhance the accuracy of the recommendations made by GroupReM. We also intent to enhance the recommendation strategy of GroupReM by considering the fact that some members of a group may influence the remaining group members in making decisions on (non-)relevant items suggested to the group.

References

[AK11] G. Adomavicius and Y. Kwon. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. In *IEEE TKDE - To Appear*, 2011.

[aUoM] GroupLens Research at University of Minnesota. Available at http://www. grouplens.org/system/files/hetrec2011-movielens-2k.zip.

[AYRC+09] S. Amer-Yahia, S. Basu Roy, A. Chawlat, G. Das, and C. Yu. Group Recommendation: Semantics and Efficiency. *VLDB Endowment*, 2(1):754–765, 2009.

[BC11] L. Boratto and S. Carta. State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups. *Studies in Computational Intelligence*, 324:1–20, 2011.

[BCC+09] L. Boratto, S. Carta, A. Chessa, M. Agelli, and M. Clemente. Group Recommendation with Automatic Identification of Users Communities. In *IEEE/WIC/ACM WI-IAT*, pages 547–550, 2009.

[BCC10] A. Bellogin, I. Cantador, and P. Castells. A Study of Heterogeneity in Recommendations for a Social Music Service. In *ACM HetRec*, pages 1–8, 2010.

[BF10] S. Berkovsky and J. Freyne. Group-Based Recipe Recommendations: Analysis of Data Aggregation Strategies. In *ACM RecSys*, pages 111–118, 2010.

[BFC09] S. Berkovsky, J. Freyne, and M. Coombe. Aggregation Trade Offs in Family Based Recommendations. In *AI*, pages 646–655, 2009.

[BMR10] L. Baltrunas, T. Makcinskas, and F. Ricci. Group Recommendations with Rank Aggregation and Collaborative Filtering. In *ACM RecSys*, pages 119–126, 2010.

[BT03] S. Banerjee and T.Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *IJCAI*, pages 205–810, 2003.

[CBH02] A. Crossen, J. Budzik, and K. Hammond. Flytrap: Intelligent Group Music Recommendation. In *ACM IUI*, pages 184–185, 2002.

[CBV10] I. Cantador, A. Bellogin, and D. Vallet. Content-based Recommendation in Social Tagging Systems. In *ACM RecSys*, pages 237–240, 2010.

[CC11] I. Cantador and P. Castells. Extracting Multilayered Communities of Interest from Semantic User Profiles: Application to Group Modeling and Hybrid Recommendations. *Computers in Human Behavior*, 27(4):1321–1336, 2011.

[CCB09] G. Cormack, C. Clarke, and S. Buettcher. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *ACM SIGIR*, pages 758–759, 2009.

[Chr08] P. Christen. Automatic Record Linkage Using Seeded Nearest Neighbor and Support Vector Machine Classification. In *ACM SIGKDD*, pages 151–159, 2008.

[CMS10] W. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison Wesley, 2010.

[CS11] I. Christensen and S. Schiaffino. Entertainment Recommender Systems for Group of Users. *Expert Systems with Applications*, 38:14127–14135, 2011.

[DK04] M. Deshpande and G. Karypis. Item-based Top-N Recommendation Algorithms. *ACM TOIS*, 22(1):143–177, 2004.

[GN08] N Gustafson and Y.-K. Ng. Augmenting Data Retrieval with Information Retrieval Techniques by Using Word Similarity. In *NLDB*, pages 163–174, 2008.

[GWB+10] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, and X. He. Document Recommendation in Social Tagging Services. In *WWW*, pages 391–400, 2010.

[GXL+10] M. Gartrell, X. Xing, Q. Lv, A. Beach, R. Han, S. Mishra, and K. Seada. Enhancing Group Recommendation by Incorporating Social Relationship Interactions. In *ACM GROUP*, pages 97–106, 2010.

[GZR+10] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel. Social Media Recommendation Based on People and Tags. In *ACM SIGIR*, pages 194–201, 2010.

[HKBR99] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *ACM SIGIR*, pages 230–237, 1999.

[JS07] A. Jameson and B. Smyth. Recommendation to Groups. In *P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.), The Adaptive Web: Methods and Strategies of Web Personalization*, pages 596–627, 2007.

[KN06] J. Koberstein and Y.-K. Ng. Using Word Clusters to Detect Similar Web Documents. In *KSEM*, pages 215–228, 2006.

[LC98] C. Leacock and M. Chodorow. *Combining Local Context and WordNet Similarity for Word Sense Identification*. The MIT Press, 1998.

[Lee97] J. Lee. Analyses of Multiple Evidence Combination. In *ACM SIGIR*, pages 267–276, 1997.

[MA98] J. McCarthy and T. Anagnost. MusicFX: An Arbiter of Group Preferences for Computed Supported Collaborative Workouts. In *CSCW*, pages 363–372, 1998.

[Mas04] J. Masthoff. Group Modeling: Selecting a Sequence of Ttelevision Items to Suit a Group of Viewers. *UMUAI*, 14(1):37–85, 2004.

[MG06] J. Masthoff and A. Gatt. In Pursuit of Satisfaction and the Prevention of Embarrassment: Affective State in Group Recommender Systems. *User Modeling User-Adapted Interaction*, 16(3-4):281–319, 2006.

[MS03] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 2003.

[MSC+06] K. McCarthy, M. Salamo, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Group Recommender Systems: A Critiquing Based Approach. In *ACM IUI*, pages 267–269, 2006.

[OCKR01] M. O'Connor, D. Cosley, J. Konstan, and J. Riedl. PolyLens: A Recommender System for Groups of Users. In *ECSCW*, pages 199–218, 2001.

[ORTN10] P. Ochi, S. Rao, L. Takayama, and C. Nass. Predictors of user perceptions of web recommender systems: How the basis for generating experience and search product recommendations affects user responses. *International Journal of Human-Computer Studies*, 68:472–482, 2010.

[PLN09] M.S. Pera, W. Lund, and Y.-K. Ng. A Sophisticated Library Search Strategy using Folksonomies and Similarity Matching. *JASIST*, 60(7):1932–1406, 2009.

[PN10] M.S. Pera and Y.-K. Ng. A Naive Bayes Classifier for Web Document Summaries Created by Using Word Similarity and Significant Factors. *IJAIT*, 19(4):465–486, 2010.

[PN11] M.S. Pera and Y.-K. Ng. With a Little Help from My Friends: Generating Personalized Book Recommendations Using Data Extracted from a Social Website. In *IEEE/WIC/ACM WI-IAT*, pages 96–99, 2011.

[PPC08] M. Park, H. Park, and S. Cho. Restaurant Recommendation for Group of People in Mobile Environments Using Probabilistic Multi-criteria Decision Making. In *APCHI*, pages 114–122, 2008.

[QPN12] R. Qumsiyeh, M.S. Pera, and Y.-K. Ng. Generating Exact- and Ranked Partially-Matched Answers to Questions in Adverstisments. In *VLDB*, page to appear, 2012.

[RAYC+10] S. Basu Roy, S. Amer-Yahia, A. Chawla, G. Das, and C. Yu. Space Efficiency in Group Recommendation. *VLDB Journal*, 19(6):877–900, 2010.

[RGJDSRDA09] J. Recio-Garcia, G. Jimenez-Diaz, A. Sanchez-Ruiz, and B. Diaz-Agudo. Personality Aware Recommendations to Groups. In *ACM RecSys*, pages 325–328, 2009.

[SWW10] Y. Shavitt, E. Weinsberg, and U. Weinsberg. Building Recommendation Systems Using Peer-to-peer Shared Content. In *ACM CIKM*, pages 1457–1460, 2010.

[YLAY09] C. Yu, L. Lakshmanan, and S. Amer-Yahia. It Takes Variety to Make a World: Diversification in Recommender Systems. In *EDBT*, pages 368–378, 2009.

[ZWZ+10] H. Zheng, D. Wang, Q. Zhang, H. Li, and T. Yang. Do Clicks Measure Recommendation Relevancy?: An Empirical User Study. In *ACM RecSys*, pages 249–252, 2010.