Performance Evaluation of Aggregation-based Group Recommender Systems for Ephemeral Groups

EDGAR CEH-VARELA, Department of Mathematical Sciences, Eastern New Mexico University
HUIPING CAO, Department of Computer Science, New Mexico State University
HADY W. LAUW, School of Information Systems, Singapore Management University

Recommender Systems (*RecSys*) provide suggestions in many decision-making processes. Given that groups of people can perform many real-world activities (e.g., a group of people attending a conference looking for a place to dine), the need for recommendations for groups has increased. A wide range of Group Recommender Systems (*GRecSys*) has been developed to aggregate individual preferences to group preferences. We analyze 175 studies related to *GRecSys*. Previous works evaluate their systems using different types of groups (sizes and cohesiveness), and most of such works focus on testing their systems using only one type of item, called Experience Goods (EG). As a consequence, it is hard to get consistent conclusions about the performance of *GRecSys*. We present the aggregation strategies and aggregation functions that *GRecSys* commonly use to aggregate group members' preferences. This study experimentally compares the performance (i.e., accuracy, ranking quality, and usefulness) using *four* metrics (Hit Ratio, Normalize Discounted Cumulative Gain, Diversity, and Coverage) of *eight* representative *RecSys* for group recommendations on ephemeral groups. Moreover, we use two different aggregation strategies, 10 different aggregation functions, and two different types of items on two types of datasets (EG and Search Goods (SG)) containing real-life datasets. The results show that the evaluation of *GRecSys* needs to use both EG and SG types of data, because the different characteristics of datasets lead to different performance. *GRecSys* using Singular Value Decomposition or Neural Collaborative Filtering methods work better than others. It is observed that the Average aggregation function is the one that produces better results.

CCS Concepts: • **Computing methodologies** → *Machine learning*; • **Information systems** → **Collaborative filtering;**

Additional Key Words and Phrases: Group recommender systems, aggregation strategies, recommendation scenarios

ACM Reference format:
Edgar Ceh-Varela, Huiping Cao, and Hady W. Lauw. 2022. Performance Evaluation of Aggregation-based Group Recommender Systems for Ephemeral Groups. *ACM Trans. Intell. Syst. Technol.* 13, 6, Article 98 (September 2022), 26 pages.
https://doi.org/10.1145/3542804

This work was partially supported by NSF Grants No. 1914635 and No. 1757207.
Authors' addresses: E. Ceh-Varela, Department of Mathematical Sciences Eastern New Mexico University 1500 S. Avenue K, Station 18 Portales, NM 88130; email: eduardo.ceh@enmu.edu; H. Cao, Department of Computer Science New Mexico State University MSC CS, P.O. Box 30001 1290 Frenger Mall SH 123 Las Cruces, NM 88003-8001; email: hcao@cs.nmsu.edu; H. W. Lauw, School of Information Systems Singapore Management University 81 Victoria Street Singapore 188065; email: hadywlauw@smu.edu.sg.

2157-6904/2022/09-ART98 $15.00
https://doi.org/10.1145/3542804

1 INTRODUCTION

Recommender Systems (*RecSys*) help users get through the problem of information overload [89]. The main task of a recommender system is to provide suggestions to users or groups of users in various decision-making processes, such as deciding the items to buy, the places to visit, or the news to read [77]. Most existing works on *RecSys* make recommendations to individual users. Increasing efforts have been put to make recommendations to groups of people [5, 18, 59].

This increase is because many real-world activities (e.g., dining at a restaurant, planning a trip) are performed in groups. For such group activities, *RecSys* have to suggest relevant recommendations using the preferences of individual group members. Systems that make recommendations for groups of users are called **Group Recommender Systems (*GRecSys*)**. *GRecSys* deal with two types of groups: persistent and ephemeral [73, 91]. This study focuses on comparing aggregation-based *RecSys* for ephemeral groups. Ephemeral groups are formed in an ad hoc manner for a specific activity and may dissolve after the activity. Group members may form the group for the first time [36, 43]. One key characteristic of ephemeral groups is the nonexistence of historical group activities [31, 69]. This makes recommendations for ephemeral groups challenging. To alleviate this issue, some studies [21, 91] attempt to relax ephemeral groups as occasional groups, in which group members have sparse historical interactions. Most *RecSys* for ephemeral groups propose aggregation strategies [76] and aggregation functions [30, 53] to combine the preferences of group members when deriving the overall group preference toward an item of interest.

After analyzing 175 works related to *GRecSys*, we observe that although significant progress has been made in this field, many challenges are not yet well addressed. One of these challenges is how *GRecSys* are evaluated [15]. Each proposed solution establishes its objectives and evaluation metrics without emphasizing comparisons with baselines considering different usage scenarios [23, 55]. This situation does not allow a fair comparison between methods, which generates many questions such as which algorithm performs better in different situations, how and when they can be used, and which solution is the best given a particular recommendation scenario (or setting).

We have identified *three specific issues* in the evaluation of *GRecSys* after a systematic review of the literature in the recent years (i.e., 2010 to 2021). First, there is no defined standard regarding the number of group members. Existing works use small groups [27, 69] medium-sized groups [34, 51], large groups [47, 66], and even very large groups [2, 55]. The conclusions of these works are sometimes contradictory. For example, some studies conclude that the performance of *GRecSys* in small groups is better [6, 15], while others get the opposite conclusion (i.e., *GRecSys* generally work worse for small groups [6, 63]).

Second, for pure ephemeral groups, to the best of our knowledge, there is no publicly available dataset with ground truth information about groups and preferences of group members. Most studies on ephemeral groups design their own strategies to create user groups. Groups can be formed (a) randomly [25, 63], (b) with high cohesion among members (i.e., similar users) [65, 67], (c) with entirely dissimilar members [23, 58], or (d) using combinations or variations of those previous methods [16, 62]. Some studies [21, 91] relax the definition of ephemeral groups and use datasets with information of occasional groups. Not all studies consider all these different types of groups when conducting their evaluations. Moreover, studies using proprietary datasets usually do not indicate how coherent (e.g., similar, dissimilar, random) the members are [68, 81].

Third, in online stores, items can be classified as **Experience Goods (EG)** and **Search Goods (SG)** [57]. For example, movies are EG because a user needs to watch the movie to perceive its attributes. Digital cameras are SG because a user can get its attributes without a previous interaction. These items have features that influence the way users prefer one item over another. The vast majority of research papers use the MovieLens dataset (http://grouplens.org/datasets/movielens/)

Table 1. Papers Inclusion Criteria for the Systematic Review

Inclusion Criteria
IC1 The paper proposes a technique to recommend items for a group of users.
IC2 The paper presents a comparison of the proposed technique and other group recommender systems.
IC3 The paper proposes techniques and metrics to evaluate recommendations for a group of users.

or similar datasets. The evaluation using different types of items with various characteristics has not been previously investigated. In summary, the *lack of consistent evaluation settings* about *GRecSys* performance concerning the size of the groups, the characteristics of group members, and the effectiveness in different types of items, raises certain doubts about in which scenarios a type of *GRecSys* is better than another or if a particular approach can be used in a different scenario.

The main goal of this study is to experimentally compare the performance (i.e., accuracy, ranking quality, and usefulness) of different aggregation-based *GRecSys* on ephemeral groups of different characteristics (i.e., size and cohesion). The major contributions of this work are as follows:

- We implement *GRecSys* based on eight different individual recommender systems. The GRecSys implements 10 aggregation functions, two aggregation strategies, four group sizes, and four types of group cohesion (Table 4). Four metrics are implemented to measure *GRecSys* performance. Different ephemeral groups are created. The implementation of the systems with the different configurations is in Reference [24] for reusable purpose.
- This is the first extensive study on utilizing datasets from two different categories, SG and EG, which have different characteristics. The datasets in Reference [24] can be used as a benchmark for future studies.
- We have conducted extensive experiments and in-depth analysis. The summarized analysis results (Section 6) can provide a guideline to future research in this direction.

The remainder of the article is organized as follows. We first present an overview of *RecSys* and *GRecSys* in Section 2. In Section 3, we present the elements from different group recommendation scenarios, including group sizes, group types, and item types. We describe the experimental settings in Section 4. The results from our tests are presented in Section 5. Section 7 concludes the article.

2 RECOMMENDER SYSTEMS

We have conducted a systematic review following [56]. We used three well-known databases (i.e., ACM, IEEE, and ScienceDirect) to find the literature related to *GRecSys*. We are interested in works published in peer-reviewed venues for the recent years (i.e., from 2010 to 2021), having the words "group" and "recommendation" (and its variations) in their titles. Three inclusion criteria (IC, Table 1) were used to select papers for our review; this allowed us to obtain 175 papers.

2.1 RecSys for Individual Users

RecSys can be classified into six categories [20]: (i) **collaborative filtering (CF)**, (ii) **content-based (CB)**, (iii) demographic, (iv) knowledge-based, (v) community-based, and (vi) hybrid. Among these categories of *RecSys*, CF, CB, and Hybrid types are the most widely utilized.

The most common CF strategies are based on *user-user* (user-based) and *item-item* (item-based) similarity [74]. The first type of CF is memory-based CF. *RecSys* from this type use ratings to calculate the similarity between all pairs of users or items. The most commonly used are **User-based CF (UBCF)** and **Item-based CF (IBCF)**. These CF approaches can also be combined in a single approach, which uses a weighted mean for the predictions of both UBCF and IBCF methods.

In this work, we refer to this combined method as IUCF. The second type of CF is model-based CF, which use a **singular value decomposition (SVD)** method [49] to decompose a sparse matrix (user matrix and item matrix) into two latent factor matrices. SVD++ [90] is an extension of the SVD algorithm.

Content-based approaches use information about items' features [14]. Based on these features, CB systems build item profiles and user preference profiles. With this information, CB systems build a model explaining the interactions between users and items. Hybrid systems combine CF with CB to address the limitations in CF or CB by using different strategies, such as weighted, mixed, switched, feature combination, and feature augmentation, to improve the accuracy of recommendations [19].

Deep Learning-based Recommender Systems have emerged combining *RecSys* from these three categories and **Deep Neural Networks (DNN)** techniques. These methods attempt to obtain low-level representations for the users and items (i.e., embeddings), whereas learning their relationships directly from the data [42, 85]. **Neural Collaborative Filtering (NCF)** [41, 48] *RecSys* is a state-of-the-art neural CF model and is considered as a generalization of Matrix Factorization models [41, 42, 48]. NCF approaches use a DNN architecture to combine user and item embeddings to capture their interactions directly from data. Recent *RecSys* methods, suited as the basis for *aggregation-based RecSys*, still use traditional approaches such as *collaborative-filtering based on similarity* [3, 8, 9, 12, 13]. Some other works on *RecSys* use modifications of conventional methods such as *matrix factorization* [28, 70] and *NCF* techniques [41, 48]. Another set of works [38, 95] applies newer techniques to utilize extra information (i.e., social relationships, tags, temporal information, user reviews, context) for making recommendations.

2.2 Recommender Systems for Groups

The *RecSys* recommending items to a group of users is called *GRecSys* [54]. Group recommendation is a complex task. First, the preferences of each group member need to be considered. Depending on the group characteristics, conflict may arise when users have different preferences toward the same item or a set of items. Second, many strategies exist to aggregate preferences of individuals. Different strategies may output different recommendations, which could impact the group members' final satisfaction.

2.2.1 Aggregation Strategies. Algorithms for *GRecSys* are in many cases based on those algorithms used in individual *RecSys*. Different aggregation strategies, derived from social choice theory, are applied to these algorithms when used for group recommendations. Two basic aggregation strategies exist for *GRecSys* [76]: (i) *individual-recommendations aggregation*, also called **aggregated predictions or aggregated ratings (PRED)** and (ii) *individual-preferences aggregation*, also called aggregated models or **aggregated profiles (PROF)**.

Most works [60, 80] use the PRED aggregation strategy, which does the aggregation in a late stage. The basic idea of this strategy is that recommendations are first made independently for each group member. Then group recommendations are produced by aggregating the ratings for each of these individual recommendations. Approaches using the PROF aggregation strategy [75, 92] creates a profile for a *"virtual"* user by aggregating each group member's item preferences, representing the preferences of the group. Therefore, the aggregation step is done first. Finally, recommendations are calculated for this *"virtual"* user.

2.2.2 Aggregation Functions. For any of the aggregation strategies mentioned before, a major concern is the generation of group recommendations considering each member's individual preferences. Based on social choice theory, several aggregation functions were presented in References [53] and [30]. Most works use some *"consensus"* functions. In Reference [94] an

Average function is implemented using the group members' rating lists. A set of works use functions focusing on those *most popular items* (i.e., majority voting). For example, Reference [96] uses the *Borda Count* function. Similarly, some aggregation functions take into account only a *subset of user preferences*. The most widely used is the *Least Misery* [34, 65] and the *Most Pleasure* [33, 65] functions. We describe the aggregation functions utilized in this study as follows:

(1) **Additive (ADD):** This function ranks items based on the addition of group members' ratings.
(2) **Approval (APP):** For each item, this function counts how many group members have rated it above a threshold. Then, items are ranked using this count.
(3) **Average (AVG):** In this function, for each item, the average of group members' ratings is calculated, then the items are ranked based on their average.
(4) **Average Without Misery (AWM):** In this function, items with ratings below a certain threshold are removed from the candidate list, then items are ranked based on their average.
(5) **Borda Count (BC):** This function first assigns a score to the items based on their ranking in the list of each user. Then, it ranks the items using their total score.
(6) **Least Misery (LM):** This function recommends the item that maximizes the minimum rating of all the group ratings.
(7) **Most Pleasure (MP):** This function measures the group's satisfaction by using the maximum group members' ratings.
(8) **Most Respected Person (MRP):** This function considers the expertise of group members toward an item. The ratings of the users who are considered experts are used for the group.
(9) **Multiplicative (MUL):** In this function, items are ranked based on the result of multiplying group members' ratings.
(10) **Popularity (POP):** This function considers those items that are the most popular.

The formal definitions of these aggregation functions can be found in our technical report [26]. For different problems and domains, some of these strategies work better than others. Although these strategies are frequently used, there is no formal study to indicate when to use one or the other or if the combination of any of these can provide better results when they are used in aggregation-based *GRecSys*. In this study, we evaluate these strategies for different scenarios.

2.2.3 Dynamic Aggregation Strategies. *GRecSys* should be able to learn users' personal preferences and model how group members reach a decision. Based on representation learning, some works have recently used **Deep Learning-based (DL)** techniques to find group preference representations dynamically. These works find the representation of a group and use that representation with techniques such as NCF [21, 37], neural preference encoders [73, 91], or **Graph Neural Networks (GNN)** [36] to make recommendations. Proposed models use the information regarding the interaction of group members and items directly from the data to learn the influence that a group member could have on a group.

An NCF framework [21, 22] is used to learn the aggregation strategy for both group recommendation and item recommendation simultaneously. The user-item and group-item interactions are learned using the same embedding space. A stacked social self-attention neural network is proposed in Reference [37] to model the interactions as a voting process among group members in groups formed ad hoc. The proposed method is enhanced with two types of aggregation methods based on item and social relationships. NCF for representation learning is also leveraged in Reference [43] to learn multi-view embeddings for the groups, users, and items using an interaction graph. Multi-view embeddings for group members are learned by incorporating the information from their inherent interests, their interacting items, and their group participation. Likewise, multi-view embeddings for items use their inherent features, and their interactions with users and groups.

Finally, group's embeddings are learned using its members' multi-view embeddings. Dynamic aggregation is done using an attention mechanism to infer the dynamic weight of each counterpart element (users, items, and groups).

Similarly, References [91] and [92] integrate an attention mechanism and a bipartite graph embedding model to leverage user-item interaction to learn and adapt each user's influences on different groups and improve recommendations. To overcome the issue of group interaction sparsity in ephemeral groups, Reference [73] uses neural preference encoders to regularize the user-group latent space dynamically prioritizing the preferences of highly informative members using different weights based on contextual preferences. To alleviate the sparsity of user-item interaction in groups formed ad hoc, a GNN representation learning network [36] is presented to improve the learning of users' preferences. The proposed approach creates an hypergraph and uses hyperedges to model groups.

2.2.4 Persistent, Ephemeral, and Occasional Groups. Recommendations are made to two types of groups based on their existing historical information: *persistent* and *ephemeral* [61, 91]. Persistent groups have a history of past activities together. Consequently, a whole group could be treated as one user [69]. Ephemeral groups [91, 92] are formed in an ad hoc manner and do not have any historical data. Preferences of ephemeral groups must be computed based on preferences for each group member. Some studies [21, 91] have used a third type of groups, *occasional groups*, as a relaxed definition for ephemeral groups. Occasional groups have historical interactions, although they are sparse. Given the lack of existing datasets about pure ephemeral groups with information about individual user preferences, group preferences for items, different group sizes and different cohesion between group members, there is a need to form synthetic ephemeral groups to test the solutions [29, 91]. In this study, these synthetic groups are considered ephemeral groups.

The scope of this study is to compare the performance aggregation-based *GRecSys* on ephemeral groups. Therefore, we are only interested in those *GRecSys* that use the aggregation strategies presented in Section 2.2.1 and the aggregation functions presented in Section 2.2.2.

3 EVALUATION SETTINGS FOR GRECSYS

In this work, we are interested in knowing how *GRecSys* perform in different scenarios (i.e., group size, group type, and type of the items to recommend). Therefore, we analyze the *experimental setup* from the papers we examined, mentioned in Section 2, to determine how these studies evaluate their proposed methods in different scenarios. We found that only 46.2% of the papers define these criteria, and in some works, the information is not complete, because they were studies using real users with no further details about the groups. The following sections describe the results obtained from analyzing these papers.

3.1 Group Formation

Both the size of the group and the cohesion of group members affect the group formation [10, 15].

3.1.1 Group Sizes. Pelaez et al. [64] found that in e-commerce platforms, the size of groups can affect the purchasing task. Their study concludes that larger groups have more challenges with group coordination for time to task completion than smaller groups. However, larger groups obtain moderately higher gains on average than smaller groups. There is not a standard about how to categorize a group size. For example, in tourism, a group size up to 10 people is considered a small group (https://arival.travel/whats-the-optimal-tour-group-size/); for academic instruction, small groups should be no more than five students [83], while in other domains, small group size is between 7 and 12 participants (https://topgolf.com/us/plan-an-event/). We analyze the literature

to understand how existing studies handle the size of their test groups. After analyzing the group size frequencies in the different studies, we categorize the groups by their size as follows:

- Small group (S): formed by 2 to 6 members.
- Medium group (M): formed by 7 to 20 members.
- Large group (L): formed by 21 to 50 members.
- Very large group (VL): formed by more than 50 members.

Using ranges for group classification has several advantages. (i) It aligns with group definitions in social sciences [78, 88]. (ii) Using this way, we can cover all the group settings in the literature we reviewed. Figure 1(a) shows the distribution of the analyzed literature using our group size categorization. We can see that most of the existing works use small groups and medium-sized groups in their tests. Only a small number of works mention that they test their solutions with groups that are larger than 20 members. (iii) It can also help get the the performance trends of those algorithms for different group sizes. In our evaluation, to test the effect of a particular group size, we present the median of the results from 100 groups formed with different numbers of members in that group range.

One noteworthy disadvantage of this group categorization method is that it is not straightforward to capture performance changes at the boundaries from one group size to another adjacent group size (i.e., from small to medium). However, the performance at the group boundaries can still be observed by forming groups with the size at the group boundaries.

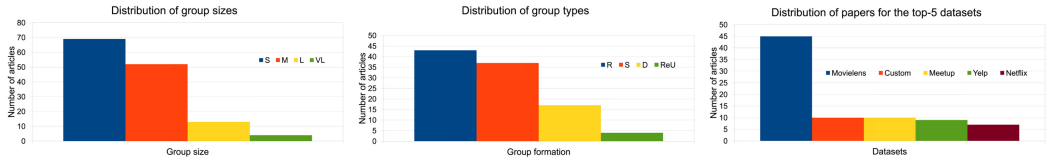
3.1.2 Group Cohesion. The way of creating groups for testing could affect the results. Studies have shown that group cohesion is a crucial aspect of group dynamics [32]. In a more cohesive group, a group member is more concerned about the opinions of other members and be willing to modify her opinions. These social and motivational forces between group members are called *group (or social) cohesion* [17].

Of all the studies that test solutions using groups with different group cohesion, only Reference [5] mention that traditional ways of generating groups are not realistic. It proposes a new way to form groups where some group members have similar preferences while other group members have different preferences. Some similar work is done in Reference [62], where groups are formed using a mid-range pairwise similarity between users or using users that are closer to a cluster centroid.

We identify four general types of group formation based on their cohesion:

- **Random (RU)** a group is formed by randomly selecting users.
- **Similar (SU)** a group is formed with users presenting high similarity values.
- **Dissimilar (DU)** a group is formed with users presenting low similarity values.
- **Realistic (ReU)** a group is formed with users having similar and dissimilar preferences.

The first three ways of group formation are the most used in the studies. However, variations on the way of forming the groups exist. For example, some works first calculate the average similarity for the whole group and then find which users are more similar or dissimilar to the average, while others use a pairwise similarity value to get the next user for a group. Figure 1(b) shows the distribution of papers using our cohesion-based group type categorization. We can observe the different approaches used by the studies. The use of random groups for testing is the most applied method. However, members of these groups shared preferences in different degrees for a given task. Therefore, in some cases, it is preferred to have groups with totally dissimilar members. For example, testing a new product before launching it to production is desirable to know a broader set of opinions. Therefore, it is of great importance to test the performance of *GRecSys* with these types of groups.



(a) Group Sizes (b) Group Cohesion (c) Datasets

Fig. 1. Distribution of papers mentioning (a) group sizes, (b) group types, and (c) distribution of top-5 datasets.

3.2 Item Types

Users' experience in e-commerce environments is also affected by the types of items on which they collect information and make a purchase decision. Items that can be bought in e-commerce websites can be classified [45] as SG and EG. Search goods are defined as items with easy search characteristics, whose quality attributes can be *inspected* before buying [84]. Examples of SG include digital cameras and clothes. Experience goods are defined as items with high search costs. Given the subjective characteristics of these items, the buyer needs to *experience* the item to determine its quality [84]. Examples of EG include music, wine, or movies. A purchase decision for *SG* may have different information requirements than a purchase decision for *EG*. Existing works on *GRecSys* do not make a detailed evaluation of how they behave when recommending different types of goods. Most of these works compare their proposals and baselines using datasets of *EG*, for example, movies, music, beer, or venues (i.e., POI). In the literature, we are only aware of one article [86] that uses *SG*. However, the paper does not clearly define what these items are.

Figure 1(c) shows the distribution of articles for the top-5 datasets used in our reviewed papers. We grouped those papers using any version of the MovieLens dataset (http://grouplens.org/datasets/movielens/). This dataset is the most widely used for testing *GRecSys*, and the items belong to the *EG* category. The second frequently used datasets are specifically designed to test *GRecSys* in studies involving real users. These datasets have specific qualities for each study and are not publicly accessible [62, 72]. The items that these datasets recommend are songs, movies, or locations. Therefore, they fall into the category of *EG*. The third and fourth frequently used datasets are used in studies recommending POIs, like restaurants (i.e., Yelp (https://www.yelp.com/dataset/challenge) and Meetup (https://www.meetup.com/es/topics/open-data/)). These datasets also belong to *EG*. Finally, the fifth most used dataset is the Netflix dataset (https://www.kaggle.com/netflix-inc/netflix-prize-data), which is similar to the MovieLens dataset and belongs to *EG*.

Studies have shown that there are important differences in the browsing and purchase behavior of consumers for these two types of goods (i.e., SG and EG) [45]. We also observe that SG datasets (from Amazon) present different characteristics (i.e., sparsity, ratings per user, and ratings per item) (see Table 3), which may affect the performance of a recommender system. However, 99.3% of the studies related to *GRecSys* evaluate their approaches using items from the *EG* category. Given these, it is necessary to evaluate *GRecSys* with other types of datasets to avoid any bias that may exist when using only a particular item type. This study analyzes the performance of aggregation-based *GRecSys* when datasets for different types of items (i.e., SG and EG) are used to know which combination of aggregation strategies and aggregation functions works the best.

4 EXPERIMENTAL SETUP

This section presents the different elements used to test *GRecSys*. POI recommendations use an additional temporal context and other contextual features, such as timestamps of user check-ins [35].

Table 2. Description of the Amazon and MovieLens Datasets

Type	Dataset	# users	# items	# reviews	avg. rating	std.
SG	TOOLS	16,638	10,217	134,476	4.36	1.03
EG	FOOD	14,681	8,713	151,254	4.24	1.09
	MOVIE	**943**	**1,682**	**100,000**	**3.53**	**1.12**

Therefore, in this study, we are not interested in POI recommendations; we are only interested in recommending items that users can get on e-commerce websites.

4.1 Datasets

We want to test if the type of items (i.e., EG and SG) influences the *GRecSys* results. Datasets for testing *GRecSys* on ephemeral groups are usually obtained from datasets used for testing *RecSys* for individual users due to the lack of publicly available datasets with ground truth information for this particular type of groups [29, 37]. Some of the existing datasets used in group recommendations contain only the group's preferences, but not the group member's preferences, making such datasets unsuitable for our study. Examples of such datasets include the Meetup dataset, used in Reference [52]; the Plancast dataset, used in Reference [93]; and the Yelp and Douban-Event datasets, used in References [37, 73]. In Reference [21] two datasets having information about the users and occasional groups are used. The first dataset, crawled from the Mafengwo website (www.mafengwo.cn), is not publicly available. The second dataset is a processed version of the CAMRa2011 dataset (http://2011.camrachallenge.com/2011). This dataset, also used in Reference [91], has an average group size of 2.08 users. We do not consider these datasets appropriate for our study, because the effect of group sizes cannot be tested. Despite this, we extend our evaluation we use the CAMRa2011 dataset to compare the performance of *GRecSys*.

We test different *GRecSys* using real-life datasets for both types of items (i.e., EG and SG). The selected datasets come from the Amazon Review dataset [40]. It contains reviews and metadata for different types of items spanning from May 1996 to July 2014. In particular, we use Tools and Home Improvement (TOOLS) as an example of SG and Grocery and Gourmet Food (FOOD) as an example of EG. Similarly, we use MovieLens-100k (MOVIE), as it is the most widely used benchmark dataset. MOVIE is also an example of EG. An analysis using more datasets can be found in our technical report [26]. Table 2 shows the description of the datasets used in this work.

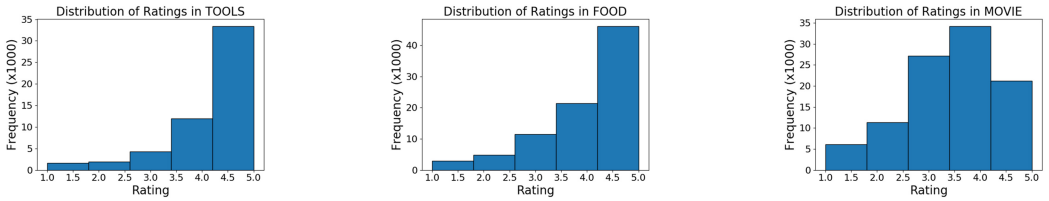
The literature states that real-life users rate a small number of items [46]. One characteristic of these real-life datasets is their sparsity [4]. A low number of ratings causes high sparsity. We can see this behavior on the Amazon datasets. Table 3 shows that the MOVIE dataset is a denser dataset, where on average, users have rated many more items than those in the Amazon datasets. The MovieLens website asks users to rate at least 15 items on the sign-up process [39]. Another difference between the MOVIE dataset and the Amazon datasets is in rating distribution. As Figure 2 shows, users in the Amazon datasets tend to give high ratings to items, whereas in the MOVIE dataset, we can observe a more diverse distribution among the rating range. These findings (i.e., rating sparsity and distribution) motivate the importance of testing *GRecSys* with real-life datasets that reflect the real users' behavior. As in other works [25, 47], we remove those users who have rated less than a threshold of items from the Amazon datasets. Our threshold is 10 items.

4.2 RecSys Setup

For different models, we choose the values of the parameters and hyperparameters utilizing a similar strategy as grid search to get reasonably good results.

Table 3. Comparing Amazon Datasets with MovieLens (MOVIE) Dataset

Dataset	Sparsity	Ratings per user	Ratings per item	Missing Ratings per user
TOOLS	99.9%	8.08	13.16	≈10,208
FOOD	99.8%	10.3	17.35	≈8,703
MOVIE	**93.6%**	**106.04**	**59.45**	**≈1,576**



(a) TOOLS (b) FOOD (c) MOVIE

Fig. 2. Dataset ratings distribution.

4.2.1 Collaborative Filtering. Five *RecSys*, *UBCF*, *IBCF*, *IUCF*, *SVD*, and *JP* are utilized. The first four methods are implemented using the Surprise framework (https://surprise.readthedocs.io/en/stable/). In particular, the first three methods use the basic *KNNWithMeans* CF algorithm implementation by setting K to be 50 and utilizing the mean ratings of each user. For *IUCF* (combination of *UBCF* and *IBCF*), we set the rating weight of *UBCF* to be 0.6 and that of *IBCF* to be 0.4. For SVD, where the framework uses a **stochastic gradient descent (SGD)** approach, we set the number of factors to be 20, and the number of iterations of SGD method to be 20 with a learning rate of 0.1. For *JP*, we have implemented a baseline method using a new similarity measure [9] that combines Jaccard similarity and Pearson correlation. We set K to 50 as in the other CF algorithms.

4.2.2 Content-based. To the best of our knowledge, no existing framework fully implements a CB model. We build a CB recommender system using the items categories and subcategories as attributes for the Amazon datasets. The TOOLS dataset has 1,266 categories and the FOOD has 512. For the MOVIE dataset, we use the 19 genres as attributes. We use Term Frequency Inverse Document Frequency on the categories and genres and the cosine similarity between the user profile and the item profile. The preference matrix contains values 1 or 0 depending on whether the user rated the item or not.

4.2.3 Hybrid. We combine the *SVD* method (as a CF approach, with rating weight 0.6) and the CB method (with rating weight 0.4). We give a higher weight to SVD given that CB algorithms suffer from overspecialization [82].

4.2.4 NCF. To implement this approach, we use the *fast.ai* framework (https://docs.fast.ai/collab.html) for collaborative filtering. As hyper-parameters, we use a learning rate of 0.01. The number of factors is set to 20 to be aligned with SVD. In Reference [42], the experiments with more than 10 epochs overfitted the NCF model. We train the model for 5 epochs to get reasonably good results and avoid overfitting.

4.3 Group Formation

An essential part of this study is group formation. According to the literature, different types of group formation have been used. We utilize approaches in different *GRecSys* [69, 93] to combine

group cohesion and group sizes. For each test, we form 100 different groups and report the median of the results. We do not use the average of the results of the 100 groups, because median is more robust to outliers than average. These groups are considered ephemeral, because no group history is available, and we only have the preferences for the individual users forming a group. The following sections describe the procedure to form these groups.

4.3.1 Group Size. We use four different group sizes (details see Section 3.1.1): S(mall), M(edium), L(arge), and VL (Very Large). This study analyzes *GRecSys* performance for different group sizes in general but not how *GRecSys* behaves when the group size monotonically increases. In other words, we are not interested in differences in *GRecSys* performance for a group of three members and a group of four members.

4.3.2 Group Cohesion. We define four group types (details see Section 3.1.2): RU, SU, DU, and ReU. The vast majority of past works use RU to form their groups. For the other types, the similarity between users is used to select users in a given group.

4.3.3 Group Creation. To form a group of size K, when the type of group is RU, we randomly select K users to form the group. For the other group types (i.e., SU, DU, and ReU), we first split all users into two clusters using the k-means algorithm. This step allows us to exploit the cluster properties, where members of each cluster present a high intra-similarity within their cluster and low inter-similarity with other cluster members.

For the SU type, we randomly select a user and get her cluster. We calculate this user's cosine similarity with the rest of the users of the same cluster and select the K-1 most similar users to this user to form the group. For the DU type, we also randomly select a user and get her cluster. We calculate this user's cosine similarity with users of the other cluster. The users from the other cluster must initially be dissimilar to the selected user, even though, using the similarity measure, we select the K-1 *least* similar users to this user to complete the group. Finally, for the ReU type, we want to simulate a more realistic scenario, where inside a group, there could be members similar to others, and at the same time, users dissimilar to others. To form the groups, we combine the two previous procedures (i.e., SU and DU). First, we randomly select a user. Then, we select her most similar users following the procedure for the SU type and her most dissimilar user follows the steps for the DU type. Then, we merge both sets of users (i.e., SU and DU) and randomly select without replacement K-1 users to complete the group.

4.4 Aggregation Strategies

We follow the two aggregation strategies presented in Section 2.2.1. The PRED strategy performs the aggregation task in a late stage. Therefore, in our implementation, for each group, we first get the set of items not rated in common by its group members. Then, depending on the selected recommender system (Section 4.2), we predict the missing rating of each item for each group member. Next, once we have the predictions for all group members, we apply the selected aggregation function (Section 2.2.2). Finally, we sort the results from the previous step in descending order, and we get as the group recommendation the top-N higher rated items.

The PROF strategy performs the aggregation strategy in an early stage. Hence, first, we aggregate the group members' initial ratings using the selected aggregation function (Section 2.2.2). The result of this step forms the preference profile for the *"virtual"* user representing the group. Second, depending on the selected recommender system (Section 4.2), for each *"virtual"* user, we predict its items' missing ratings. Finally, we get as the group recommendation the top-N higher rated items. As the group recommendation list for both strategies, we select the top 100 items

Table 4. Summary of Elements' Acronyms

Component	Element Acronyms
Datasets	TOOLS, FOOD, MOVIE, CAMRa
Item types	SG, EG
Group sizes	S, M, L, VL
Group cohesion	RU, SU, DU, ReU
Recommeder Systems	UBCF, IBCF, IUCF, SVD, CB, HYBRID, NCF, JP
Aggregation strategies	PRED, PROF
Aggregation functions	ADD, APP, AVG, AWM, BC, LM, MP, MRP, MUL, POP
Metrics	HR, nDCG@k, Diversity, Coverage

not rated previously by any group members. We implement the aggregation functions presented in Section 2.2.2 for both aggregation strategies. We modify some parameters for the following aggregation functions:

(1) APP: For both strategies, we use a threshold of 3.0. We consider that items with ratings close to 4 or 5 are more to the liking of users.
(2) AWM: For both strategies, we use a threshold of 2.0. We consider that items with low ratings are not of users' interest.
(3) POP: Considering the average of ratings per item for the datasets, we use a threshold of 20 ratings.

4.5 Metrics

When recommending items to users, it is essential to consider different performance metrics. To evaluate the result from the aggregation strategies and functions applied to each recommender system approach, we use metrics for *accuracy* [7], *ranking quality* [71], and *usefulness* [44].

4.5.1 *Accuracy Metrics.*

Hit Ratio (HR). This metric calculates the average percentage of items in the group recommendation list present in the individual group member recommendation list [29].

4.5.2 *Ranking Quality Metrics.*

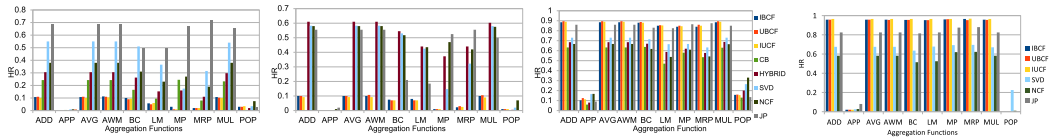
nDCG@k. We adopt the **Normalize Discounted Cumulative Gain (nDCG)** metric [87] to measure the ranking quality of each group's list of recommended items. Relevant items should appear higher in the recommendation list [11]. This metric is the most popular among all the 175 works analyzed.

4.5.3 *Usefulness Metrics.*

Diversity. Diversity measures how different the items are in a recommendation list. Having diversification in the recommendation list increases the quality of the user's experience, because it reduces the recommender system over-specialization [50].

Coverage. This metric represents the percentage of items in the recommendation list that a recommender system can recommend among the number of potential items (i.e., catalog). A higher coverage may benefit users by exposing them to a broader range of recommended items, which could increase satisfaction with the recommender system [1].

To end this section, we present Table 4, which shows the elements' acronyms for the different defined components we are going to use for our evaluations.



(a) TOOLS (SG) (b) FOOD (EG) (c) MOVIE (d) CAMRa

Fig. 3. HR metric for all recommender systems using PRED aggregation strategy.

5 PERFORMANCE EVALUATION

Several works show that no single recommender system approach is better than others for all scenarios [23]. We show results from the Amazon datasets (i.e., TOOLS and FOOD). These datasets represent a real-life environment where many users have only rated a small percentage of items. Recall that TOOLS and FOOD are SG and EG, respectively. We also present the results for MOVIE as a comparison baseline.

5.1 Occasional Groups

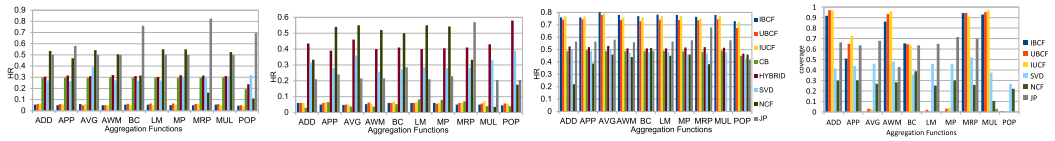
Although this work aims for the evaluation of *GRecSys* on ephemeral groups, we would like to know how these *GRecSys* work with a dataset of occasional groups. We use the CAMRa2011 dataset (CAMRa) used in Reference [21]. This dataset contains 290 groups with an average of 2.08 members per group. It contains ratings from users and groups on a scale from 0 to 100. These ratings are transformed to a new range from 0 to 5. Due to the lack of additional information for this dataset and the small size of the groups, not all tests are possible. We can only test for group size S and we assume the group cohesion is random (RU). Since this dataset does not contain information about categories or genres we cannot evaluate the *GRecSys* using the Diversity metric.

5.2 Effect of Different Aggregation Functions and Aggregation Strategies

This section analyzes how aggregation functions influence the group recommendations when different aggregation strategies are used. We fix the group size to *S(mall)* and the group cohesion to *RU*.

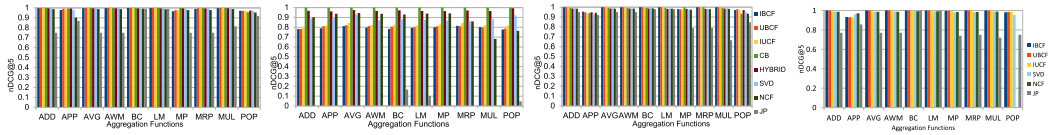
5.2.1 Hit Ratio. The HR measures the degree that the items recommended for the group match the items recommended for each group member. Figure 3 shows the HR results when the aggregation strategy is PRED. For SG items, JP produces the best results, which is followed by SVD (Figure 3(a)) for most aggregation functions. For EG items, Figure 3(b) shows that the approach with the best results is HYBRID when it is combined with ADD, AVG, AWM, BC, LM, and MUL. For MOVIE, Figure 3(c) shows that the best approaches for most of the aggregation functions are UBCF, IBCF, IUCF, and JP. These results make sense if we account that MOVIE is less sparse than TOOLS and FOOD, indicating that each user and each item has more similar neighbors with similar ratings.

Figure 4 shows the results for HR when the aggregation strategy is PROF. On SG items (Figure 4(a)), JP performs the best, which is followed by NCF. This is because JP is finding better similarities among users. On EG items (Figure 4(b)), NCF outperforms other methods for most aggregation functions. Again, the behavior of UBCF, IBCF, and IUCF is similar to the use of the PRED aggregation strategy. On the MOVIE dataset, JP is performing worse when using PROF than using PRED, this is mostly because the different user profiles that are combined in the creation of the "virtual" user (using PROF) affect the similarity with other users. For the CAMRa dataset (Figures 3(d) and 4(d)), IBCF and UBCF obtain the best results for most aggregation functions. A reason could be the high amount of previous interactions from the groups in this dataset.



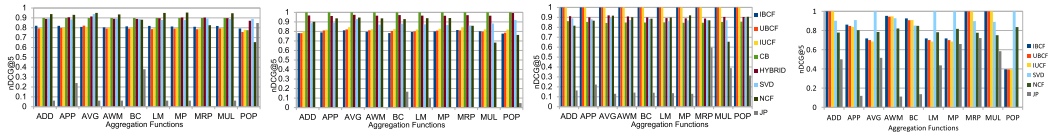
(a) TOOLS (SG) (b) FOOD (EG) (c) MOVIE (d) CAMRa

Fig. 4. HR metric for all recommender systems using PROF aggregation strategy.



(a) TOOLS (SG) (b) FOOD (EG) (c) MOVIE (d) CAMRa

Fig. 5. nDCG@5 metric for all recommender systems using PRED aggregation strategy.



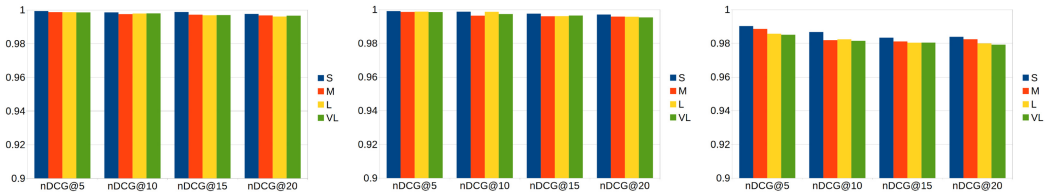
(a) TOOLS (SG) (b) FOOD (EG) (c) MOVIE (d) CAMRa

Fig. 6. nDCG@5 metric for all recommender systems using PROF aggregation strategy.

Overall, higher HR values are obtained using the PRED aggregation strategy. These results indicate that the aggregation strategy is determinant for high accuracy results for the different aggregation functions. The leading cause could be that, in PRED, individual rating predictions are aggregated in a later stage, which is more helpful as more information of the group members is used.

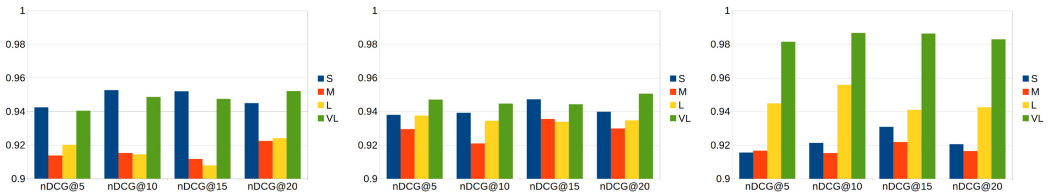
5.2.2 nDCG@k. This section analyzes to what degree the ranking of items for the group recommendation matches the ranking of items for each group member. Figure 5 presents the result of the PRED strategy. Figure 5(a) and (b) show that for both SG and EG, most methods get similarly good results with most of the aggregation functions. JP is the worst performing strategy, because the ratings it generated for the recommended items are not highly ranked on each group member's ranking. Figure 6(a) and (b) show that for both SG and EG items, CB, HYBRID, SVD, and NCF have the best results when using the PROF aggregation strategy combined with most aggregation functions. Figures 5(c) and 6(c) show that for MOVIE, in general, the best recommender systems are IBCF, UBCF, and IUCF (using any aggregation function). Higher values in nDCG are obtained when using the PRED aggregation strategy. For the CAMRa dataset, IBCF and UBCF presents the best results with most aggregation functions (Figure 5(d)). Figure 6(d) shows that SVD can obtain better values on more aggregation functions. Given relatively rich group interactions in this dataset, the SVD method with the PROF aggregation strategy seems to better use the similar profiles within the groups.

Results from this section indicates that the distribution of ratings (see Figure 2) in the dataset is an important factor for the high nDCG values for the different aggregation functions. As the distribution of ratings in real-life datasets is skewed toward the highest ratings (i.e., 4 and mainly 5), it is expected that the ratings in the first places of the recommendation list are high.



(a) TOOLS (SG) (b) FOOD (EG) (c) MOVIE

Fig. 7. nDCG results varying k using PRED aggregation strategy.



(a) TOOLS (SG) (b) FOOD (EG) (c) MOVIE

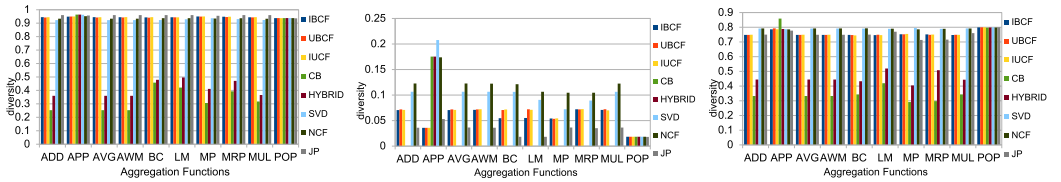
Fig. 8. nDCG results varying k using PROF aggregation strategy.

We also test the performance of the recommender systems with different k values for *nDCG@k*. Figures 7 and 8 show the nDCG results when using a wider range of values for k for the PRED and PROF aggregation strategies, respectively. While varying the group size, we fix the group to RU (i.e., random), aggregation function to AVG, and SVD and NCF for PRED and PROF. Finally, we repeated the experiments three times, and we present the average of the results. In Figure 7, we can see a clear trend in the results using the PRED aggregation strategy where smaller groups have better performance than larger ones for different values of k. However, this trend is not found when using the PROF aggregation strategy, as Figure 8 shows. First, we can see that overall, the results obtained are lower than those using the PRED aggregation strategy. Second, overall, we can see that the largest groups (i.e., VL) have better results than smaller ones. This result could indicate that as more members the group has, the virtual user profile created using the PROF aggregation strategy is more similar to the group members' preferences than using just the ratings as in PRED.

5.2.3 Diversity. Figure 9 shows the results of the PRED aggregation strategy. Less diverse recommendations are generated for the FOOD dataset (EG) than for the TOOLS dataset (SG) (Figure 9(a) and b)). We attribute this behavior to two aspects. First, items in FOOD are not distributed among many categories like TOOLS (Section 4.2.2). Second, the number of items with similar ratings in FOOD is higher than in the TOOLS dataset (Figure 2). Third, TOOLS is a very sparse dataset, affecting the rating predictions as items do not have many neighbors, then neighbors with different characteristics are used. Figure 9(a) shows that for the TOOLS dataset, JP with most of the aggregation functions. Figure 9(b) shows that NCF, with most aggregation functions, provides the best results on the FOOD dataset, even when all values are much lower. Only when using the APP aggregation function, SVD shows the best diversity score.

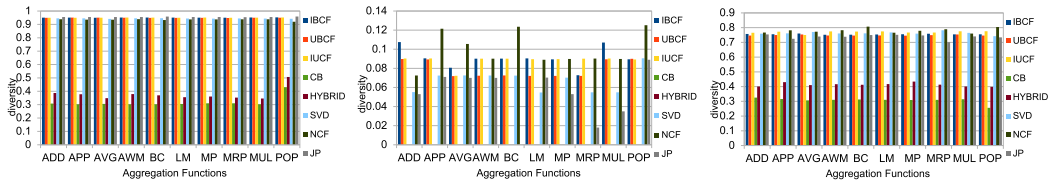
Figure 10 shows the results of the PROF aggregation strategy. We can see the same trend of low diversity values for the FOOD dataset (see Figure 10(b)). We attribute this behavior of low values to the same reasons mentioned for the PRED aggregation strategy.

Figure 10(a) shows that higher diversity values for TOOLS are obtained using JP with most of the aggregation functions. This result possibly appears because of the dataset sparsity, and therefore



(a) TOOLS (b) FOOD (c) MOVIE

Fig. 9. Diversity metric for all recommender systems using PRED aggregation strategy.



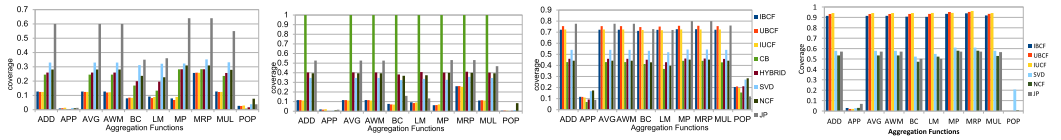
(a) TOOLS (b) FOOD (c) MOVIE

Fig. 10. Diversity metric for all recommender systems using PROF aggregation strategy.

neighbors with different features have to be used. Figure 10(b) shows that the approach showing better results is NCF combined with APP, AVG, BC, MP, MRP, and POP.

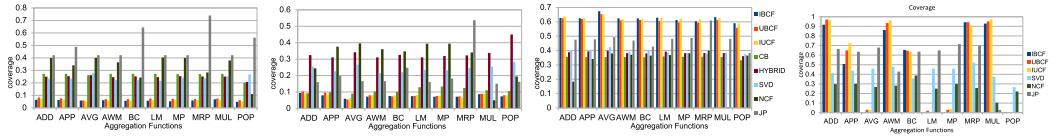
Figures 9(c) and 10(c) show that the best approach is NCF when using the MOVIE dataset. Recall that ratings are well distributed in this dataset, and items belong to a small number of genres. These dataset properties could lead NCF to find better rating patterns. The results indicate that when looking for high Diversity values, the aggregation function is influenced by the aggregation strategy and the distribution of items concerning their features (i.e., categories, genres). The number of categories seems to be the most important aspect in combination with the PRED aggregation strategy. One possible explanation is that there is no over-specialization in the recommendations as the different group member profiles are aggregated at a later stage, leading to a more diverse recommendation list.

5.2.4 Coverage. Figure 11 shows the results when using the PRED aggregation strategy. Figure 11(a) shows that the JP approach, combined with most aggregation functions, obtains the best results. However, Figure 11(b) shows a high Coverage for FOOD when using CB with almost all the aggregation functions. Only with APP and POP aggregation functions, CB's values are low. As in Diversity, we attribute this behavior to the two characteristics of FOOD. Moreover, as CB recommends items based on their features (i.e., categories or genres), the recommendation list created by most of the aggregation functions covers the catalog of items. Figure 12 shows the results when using the PROF aggregation strategy. Figure 12(a) shows that for the TOOLS dataset, JP presents the best results in combination with most of the aggregation functions. This figure also shows that the HYBRID method has high values for ADD, MUL, and POP. HYBRID results are probably dependent on SVD, which is also high for these aggregation functions, boosted with CB given that the aggregation step happens earlier in the recommendation process. Figures 11(c) and 12(c) show that when using the MOVIE dataset, the best approaches are UBCF, IBCF, and IUCF when using any aggregation functions. These results indicate that the influence of similar neighbors is strong in this dataset. When using the CAMRa dataset (Figures 11(d) and 12(d)), IUCF, UBCF, and IBCF present similarly higher results for most of the aggregation functions. This results could be related to the number of interactions the groups have in this dataset.



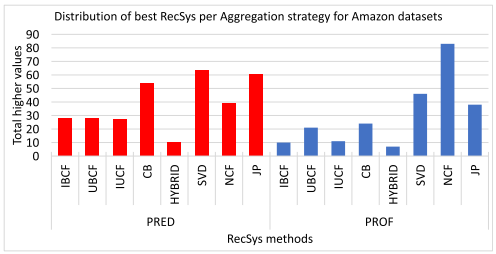
(a) TOOLS (b) FOOD (c) MOVIE (d) CAMRa

Fig. 11. Coverage metric for all recommender systems using PRED aggregation strategy.

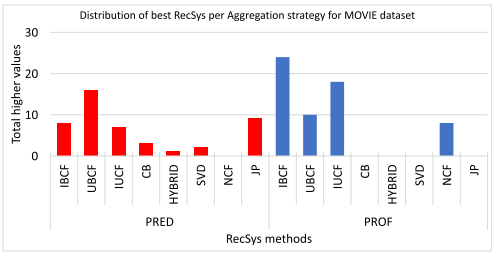


(a) TOOLS (b) FOOD (c) MOVIE (d) CAMRa

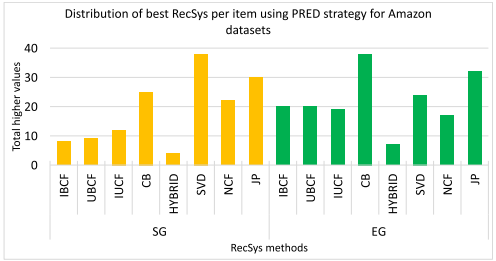
Fig. 12. Coverage metric for all recommender systems using PROF aggregation strategy.



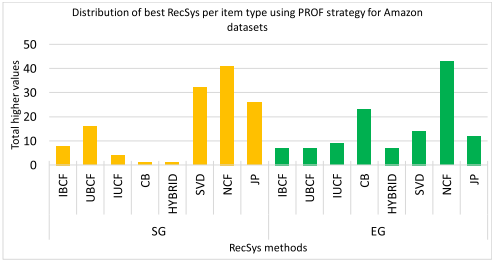
(a) PRED vs. PROF (Amazon datasets)



(b) PRED vs. PROF (MOVIE dataset)



(c) SG vs. EG (PRED)



(d) SG vs. EG (PROF)

Fig. 13. Distributions of best RecSys methods for different aggregation strategies and item types.

Results indicate that when looking for high Coverage values, the aggregation function is influenced by different factors such as the aggregation strategy, the sparsity in ratings, and the distribution of items regarding their features (i.e., categories, genres). Higher Coverage values are obtained using the PROF aggregation strategy. This observation indicates that recommendations for the "virtual" user are aligned with the group member's recommendations. Similarly, results show that in most cases, the aggregation functions ADD, AVG, AWM, and MUL produce better results with the majority of the recommendation approaches.

5.2.5 RecSys and Aggregation Function Selection. This section examines which recommender system works best for a specific aggregation strategy (i.e., PRED and PROF) and a type of items (i.e., SG and EG). We count the frequencies of tests in which a recommender system method obtains the best performance.

We first examine which methods "win" for different aggregation strategies (PRED and PROF). Figure 13(a) and (b) show the results. For Figure 13(a), 240 test results have been collected for each aggregation strategy. They correspond to the four metrics results on 60 experiments (using the 10 aggregation functions on all the six Amazon datasets (more details see Reference [26]). For Figure 13(b), 40 test results have been collected. They are the values of the four metrics on 10 experiments (using the 10 aggregation functions on the MOVIE dataset). When recommender system methods have ties, they all be counted as winning. Figure 13(a) shows that for PRED, the SVD approach has the highest value, followed by JP. For the PROF aggregation strategy, the NCF approach is the one with the highest value. For the MOVIE dataset (Figure 13(b)), the best algorithm for the PRED strategy (in red) is UBCF, while for the PROF strategy (in blue), IBCF wins. The different algorithm performance between the two types of datasets can be attributed to the number of ratings each user has made on average and also the number of ratings each item has on average. As Table 3 shows, while for Amazon datasets, these are low numbers, in the MOVIE dataset, these numbers are considerably higher. We further investigate the *winning* methods for recommending different types of items (SG and EG). Figure 13(c) and (d) show the results for the SG type (in color orange) and for the EG type (in color green), respectively. For both figures, 120 tests results are collected for each type of item, and the ties are processed using the same strategy for Figures 13(a) and (b). Figure 13(c) shows that when the aggregation strategy is PRED, for the SG type, the clear winner is the SVD algorithm, whereas, for the EG type, the CB approach is better than the SVD. However, Figure 13(d) shows that when the aggregation method is PROF, for both item types (i.e., SG (in orange) and EG (in green)), the NCF approach has the highest frequency value. Regarding the *aggregation function selection*, the AVG function is similar to making decisions in a group of people [53] when people are willing to change their preferences to reach a group agreement. Moreover, the results in previous sections show that the AVG function has better performance than the AWM function. Therefore, this is an additional reason to choosing the AVG function as the most optimal function for use in *GRecSys*.

5.3 Effect of Group Formation

This section examines how the group formation (sizes and cohesion) affects the performance of recommender approaches. We measure the performance using the four metrics. We use SVD and NCF methods for the PRED and PROF aggregation strategies, respectively. We use AVG as the aggregation function. These are the suggested effective methods and functions as shown in analysis in Section 5.2.5.

5.3.1 Effect of Different Group Sizes. For these tests, we fix the group cohesion to be RU. Using the PRED strategy, Figure 14 shows that, on the three datasets, the best HR and Coverage scores are obtained with group size S, while the worst score is from the VL groups. For nDCG and Diversity, there is no significant difference in the group sizes. The results indicate that there is a better agreement in small groups than in larger ones. Similar results can be observed from tests on other SG and EG datasets [26].

Using the PROF strategy (Figure 15), a similar observation can be made on HR and Coverage as with the PRED strategy. Regarding nDCG, there is not much difference. For Diversity, we see a change, where VL groups obtain the best values. For the FOOD and MOVIE datasets, we find that the VL groups have better nDCG values than S groups. This is mainly because the "virtual" user profile must cover more items, helping the NCF method find useful patterns. For Diversity, for these two datasets, the best results are obtained with the group size S. When the size of a group grows, the number of individual preferences that must be considered during the recommendation process also grows. Therefore, recommending new and appealing items for all members becomes more difficult.

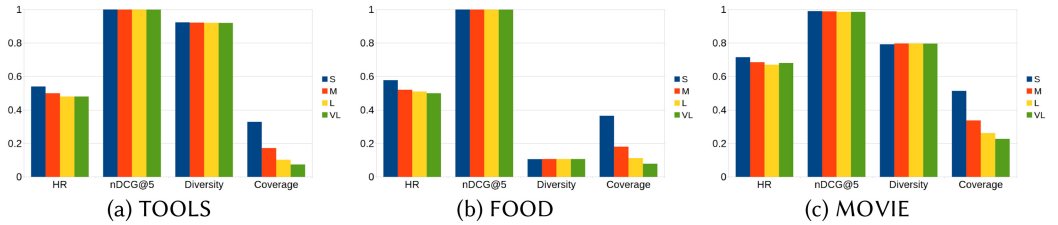


Fig. 14. Metric results for different group sizes using SVD method and PRED aggregation strategy.

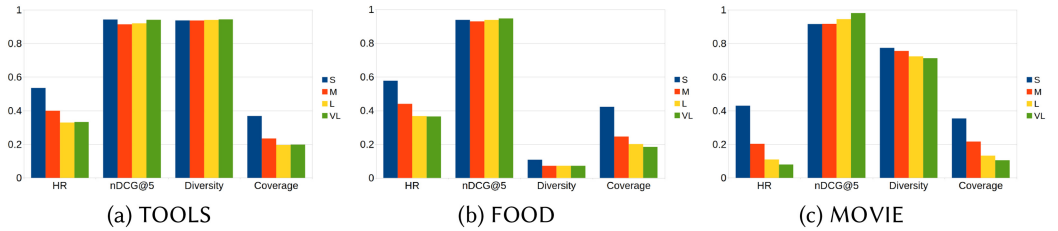


Fig. 15. Metric results for different group sizes using NCF method and PROF aggregation strategy.

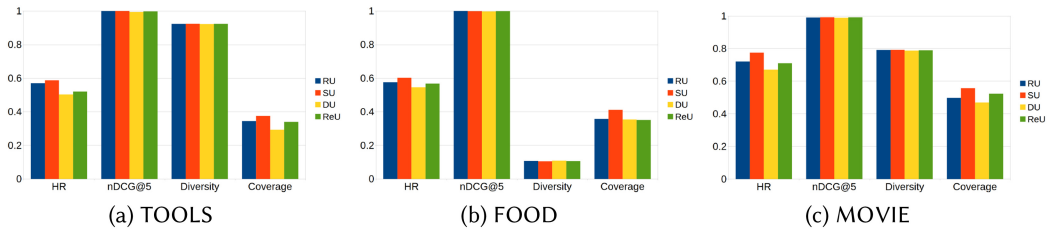
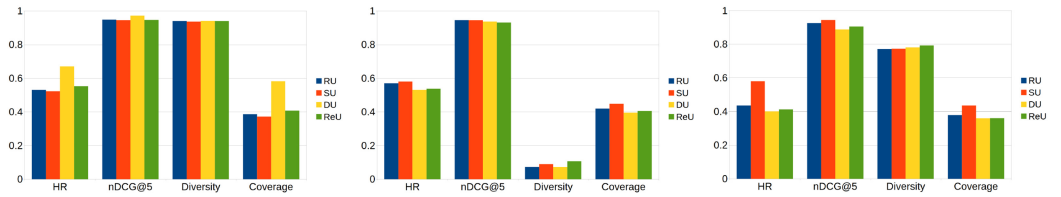


Fig. 16. Metric results for different group types using SVD method and PRED aggregation strategy.

This finding aligns well with the literature. Moreover, results using the PROF aggregation strategy (i.e., Figure 15) show more notable differences between group sizes than those using the PRED aggregation strategy (i.e., Figure 14) due to the creation of a broader profile for a group.

5.3.2 Effect of Different Group Cohesion. For these tests, we fix the group size to be *S(mall)*. Figure 16 presents the results for the PRED aggregation strategy using the SVD method. For the HR and Coverage metrics on the three datasets (i.e., TOOLS, FOOD, and MOVIE), we can see that the highest values are obtained when the group is formed by similar users (i.e., SU). Correspondingly, the lowest values are obtained when the group has dissimilar users (i.e., DU). These results make sense, since the profiles of group members should be more related in SU and more diverse in DU. For both the nDCG and Diversity metrics, there is not much difference in the results for the different group cohesion. The Diversity values for the FOOD dataset are lower than for the other datasets. This behavior is aligned with the results of Section 5.2.3, as the FOOD dataset is not distributed in many categories, and its number of items with similar ratings is higher compared with TOOLS. These results indicate that as more like-minded the group members are (i.e., SU), the group becomes more homogeneous. Therefore, the ratings these similar users give to items should also be similar.

Figure 17 presents the results for the PROF aggregation strategy using the NCF method. Figure 17(a) shows that the DU type has the highest value for HR, nDCG, and Coverage metrics. For Diversity, there is not a clear dominant group type. Consistent results can be observed



(a) TOOLS (b) FOOD (c) MOVIE

Fig. 17. Metric results for different group types using NCF method and PROF aggregation strategy.

from experiments on other SG datasets [26]. One possible explanation for these outcomes is that when the profile aggregation is done at an early stage in the group recommendation process, the *"virtual"* user profile is broader, which can be similar to more users. Figure 17(b) shows that the pattern of the results is similar to the results for the MOVIE dataset (Figure 17(c)). The Diversity values for this dataset is lower than for the other datasets. This behavior is aligned with the results of Section 5.2.3.

The results align with the literature for the PRED strategy for both types of items. However, we can see that DU groups present better results for the SG items when using the PROF aggregation strategy. These results show that when using different aggregation strategies and different types of items, there is a variation in the outcomes for the different group cohesion.

5.4 Statistical Comparison

This section utilizes statistical tests to analyze whether the choices of (a) aggregation strategies and (b) types of items make a difference when making recommendations. As in previous sections, we use the AVG aggregation function for all tests; SVD and NCF methods are chosen for the PRED and PROF aggregation strategies. The *t-test* analysis [79] is used.

5.4.1 Choice of Aggregation Strategies. The first set of statistical tests compares the effect of both aggregation strategies (PRED and PROF) by testing all *group sizes* while fixing the group cohesion to be RU. For each aggregation strategy and each metric (i.e., HR, nDCG, Diversity, Coverage), 400 results are collected from experiments on 400 groups (i.e., 100 groups for each group size, four group sizes S, M, L, VL). Table 5 shows the results. We evaluate whether there is significant difference with different aggregation strategies. We can see that there is no difference, except HR in TOOLS ($p > 0.05$), between both aggregation strategies. These results indicate that we should expect similar HR results using any aggregation strategy. The second set of statistical tests compares the PRED and PROF aggregation strategies' effect by testing all *group cohesions* while fixing group size to be S. Similar to the previous tests, we collect 400 results using the group cohesion (i.e., RU, SU, DU, and ReU). Table 6 shows the results. We evaluate each dataset to see if there is any significant difference when different aggregation strategy is used for the group types. Only for HR in both datasets, there is no difference ($p > 0.05$) between both aggregation strategies.

5.4.2 Choice of Item Types. This group of tests shows whether recommendation results are different where different types of items are used. We use the TOOLS dataset and the benchmark MOVIE dataset. This is mainly because the TOOLS dataset, a representative of the Amazon datasets, contains SG items which are less often used, while the MOVIE dataset having EG items, is the preferred benchmark dataset in most *RS*.

The first test compares the effect of using datasets with different types of items by testing all *group sizes* while fixing the group cohesion to be RU on each aggregation strategy (i.e., PRED and PROF). For each dataset and each metric, 400 results are collected from experiments on 400 user

Table 5. Statistical t-test Comparing the Effect of Two Aggregation Strategies (PRED, PROF) by Aggregating Different Group Sizes

	TOOLS (PRED, PROF)		FOOD (PRED, PROF)	
	t-stat	p-value	t-stat	p-value
HR	−1.816	$0.071 > 0.05$	−2.318	$0.021 < 0.05$
nDCG	20.777	$0.00 < 0.05$	15.762	$0.00 < 0.05$
Diversity	−13.558	$0.00 < 0.05$	−2.76	$0.006 < 0.05$
Coverage	−4.101	$0.00 < 0.05$	−3.682	$0.00 < 0.05$

Table 6. Statistical t-test Comparing the Effect of Two Aggregation Strategies (PRED, PROF) by Aggregating Different Group Cohesion

	TOOLS (PRED, PROF)		FOOD (PRED, PROF)	
	t-stat	p-value	t-stat	p-value
HR	−0.119	$0.906 > 0.05$	−1.271	$0.205 > 0.05$
nDCG	16.783	$0.00 < 0.05$	19.262	$0.00 < 0.05$
Diversity	−12.268	$0.00 < 0.05$	3.12	$0.002 < 0.05$
Coverage	−2.619	$0.009 < 0.05$	−3.173	$0.002 < 0.05$

Table 7. Statistical t-test Comparing the Metrics Obtained on Different Item Types (SG, EG) for the Two Aggregation Strategies (PRED, PROF) for Group Sizes

	PRED (TOOLS, MOVIE)		PROF (TOOLS, MOVIE)	
	t-stat	p-value	t-stat	p-value
HR	−12.512	$0.00 < 0.05$	4.572	$0.00 < 0.05$
nDCG	9.313	$0.00 < 0.05$	5.831	$0.00 < 0.05$
Diversity	62.745	$0.00 < 0.05$	43.903	$0.00 < 0.05$
Coverage	−12.628	$0.00 < 0.05$	0.171	$0.856 > 0.05$

Table 8. Statistical t-test Comparing the Metrics Obtained on Different Item Types (SG, EG) for the Two Aggregation Strategies (PRED, PROF) for Group Cohesion

	PRED (TOOLS, MOVIE)		PROF (TOOLS, MOVIE)	
	t-stat	p-value	t-stat	p-value
HR	−11.689	$0.00 < 0.05$	4.218	$0.00 < 0.05$
nDCG	11.256	$0.00 < 0.05$	4.551	$0.00 < 0.05$
Diversity	67.24	$0.00 < 0.05$	47.882	$0.00 < 0.05$
Coverage	−10.585	$0.00 < 0.05$	0.03	$0.976 > 0.05$

groups (100 groups for a specific group size (i.e., S, M, L, VL)). Table 7 shows the results of the statistical t-tests. We observe that, only for Coverage for PROF aggregation strategy, there is no difference ($p > 0.05$) between both datasets.

The second test compares the effect of different types of datasets by testing all *group cohesion types* while fixing the group size to be S on each aggregation strategy. Like in the previous tests, we collected 400 results using the four group types (RU, SU, DU, and ReU). Table 8 shows the results of the statistical t-tests. We can see that there is no difference in Coverage ($p > 0.05$) between both datasets using the PROF aggregation strategy.

5.4.3 Statistical Analysis Summary. The results show that there are evident differences between *GRecSys* using different aggregation strategies. It suggests that performing the aggregation earlier or later in the recommendation process influences the final results. These results also indicate that there is a difference between the results using different types of items. Based on these results, we can conclude that the recommendation methods show different behaviors on the Amazon datasets and the benchmark MOVIE dataset.

6 DISCUSSIONS

Our comprehensive experimental analysis has revealed several interesting and valuable insights, which can be used to guide the evaluation of future *GRecSys*. First, the type of items (i.e., SG and EG) influences the results obtained from *GRecSys*. There is a clear difference between the results obtained from datasets with SG items and datasets containing EG items. These differences suggest that it is *insufficient* to test the performance of a group recommender system using the benchmark MOVIE dataset alone. Instead, it is important to use datasets of both item types to

conduct performance analysis. On the one hand, as mentioned in Section 3.2, the rating of EG items involves a personal preference. Therefore the rating for these type of items is somehow subjective. On the other hand, the rating of items of type SG is more objective. Users can rate these items just based on the features and characteristics without involving the experience of using the item. This difference in the way users rate these items clearly affects the performance of the *RecSys*.

About group formation, we have two findings. (i) Recommendations become more difficult as the group grows. Users in larger groups have more difficulty reaching a consensus. This finding is consistent with the literature. (ii) When different types cohesion are used, we found a clear difference in the recommendation results when different aggregation strategies are used in these user groups. For the PRED aggregation strategy, the results align with the literature, where better results are obtained for groups of high cohesion, like groups of type SU. However, for the SG items when using the PROF aggregation strategy groups of type DU present better results. Regarding aggregation strategies to use, if the dataset is about items of type SG, then the best aggregation strategy to use is PRED (i.e., aggregated predictions). If we are dealing with items of type EG, then the aggregation strategy PROF (i.e., aggregated profiles) is more suitable. These finding are also based on the way users rate these items, based on the experience using the item or based on the item's features, as stated previously. For both aggregation strategies, AVG presents better performance than other aggregation functions, and it is aligned with previous works as stated in Section 5.2.5. Concerning the selection of individual recommender algorithm to use in *GRecSys*, we recommend SVD if the PRED aggregation strategy is selected, because it gets better results. If the PROF aggregation strategy is implemented, then we recommend using NCF, because it performs better. For occasional groups with existing parse previous interactions, CF method is recommended to be used because of its superior performance.

7 CONCLUSIONS AND FUTURE WORK

This paper evaluates the performance of aggregation-based *GRecSys* when used in ephemeral groups. We conducted detailed experimental comparisons of *GRecSys* by considering different factors, including the types of items to be recommended (SG or EG), the formation of groups (sizes and types), the aggregation strategies (PRED and PROF), and aggregation functions. Despite that this study focuses on recommendations to ephemeral groups, we still tested the *GRecSys* using a real-life dataset containing occasional groups, which has a relaxed definition of ephemeral groups.

As far as we know, this work is the first attempt to extensively test *GRecSys* using both SG and EG. The findings of our study can provide a basic guide for the future evaluation of *GRecSys*. Our comprehensive implementation can be used as comparison baselines. In the literature, 99.3% of the studies evaluated their approach using EG items, where the MovieLens (MOVIE) dataset is the most used. However, real-life datasets from Amazon present different characteristics (i.e., sparsity, ratings per user, and ratings per item) from the MOVIE dataset.

This study, although comprehensive, has some limitations. First, not all datasets contain the ground truth information about the groups, their members, and their interactions with different items. Therefore, as we do not have a history of interactions, all the groups formed for this study are synthetic and ephemeral. Second, we are not covering in this research the recommendation of POIs for groups. Recommending these types of items involves a temporal component for the group formation. The datasets used for this study do not have this component. Third, this study does not compare those *GRecSys* using DL models to aggregate groups and user preferences directly from data. We plan to address these limitations in future works. Specifically, we want to compare the performance of those *GRecSys* that use DL for aggregating preferences, such as those mentioned in Section 2.2.3. In addition, we are interested in exploring more extensive the group's characteristics to generate more guidelines for selecting the best *GRecSys* to use for recommendations.

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2011. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* 24, 5 (2011), 896–911.

[2] Akshita Agarwal, Manajit Chakraborty, and C. Ravindranath Chowdary. 2017. Does order matter? Effect of order in group recommendation. *Expert Syst. Appl.* 82 (2017), 115–127.

[3] Rishabh Ahuja, Arun Solanki, and Anand Nayyar. 2019. Movie recommender system using k-means clustering and k-nearest neighbor. In *Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence'19)*. IEEE, 263–268.

[4] Bushra Alhijawi, Ghazi Al-Naymat, Nadim Obeid, and Arafat Awajan. 2019. Mitigating the effect of data sparsity: A case study on collaborative filtering recommender system. In *Proceedings of the 2nd International Conference on new Trends in Computing Sciences (ICTCS'19)*. IEEE, 1–6.

[5] Irfan Ali and Sang-Wook Kim. 2015. An effective approach to group recommendation based on belief propagation. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, 1148–1153.

[6] Irfan Ali and Sang-Wook Kim. 2015. Group recommendations: Approaches and evaluation. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*. 1–6.

[7] Areej Alsini, Du Q. Huynh, and Amitava Datta. 2020. Hit ratio: An evaluation metric for hashtag recommendation. arXiv:2010.01258. Retrieved from https://arxiv.org/abs/2010.01258.

[8] Ali A. Amer, Hassan I. Abdalla, and Loc Nguyen. 2021. Enhancing recommendation systems performance using highly-effective similarity measures. *Knowl.-Bas. Syst.* 217 (2021), 106842.

[9] Ali A. Amer and Loc Nguyen. 2021. Combinations of jaccard with numerical measures for collaborative filtering enhancement: Current work and future proposal. arXiv:2111.12202. Retrieved from https://arxiv.org/abs/2111.12202.

[10] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. 2009. Group recommendation: Semantics and efficiency. *Proc. VLDB Endow.* 2, 1 (2009), 754–765.

[11] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern Information Retrieval*, Vol. 463. ACM Press, New York, NY.

[12] Sujoy Bag, Abhijeet Ghadge, and Manoj Kumar Tiwari. 2019. An integrated recommender system for improved accuracy and aggregate diversity. *Comput. Industr. Eng.* 130 (2019), 187–197.

[13] Sujoy Bag, Sri Krishna Kumar, and Manoj Kumar Tiwari. 2019. An efficient recommendation generation using relevant Jaccard similarity. *Inf. Sci.* 483 (2019), 53–64.

[14] Marko Balabanović and Yoav Shoham. 1997. Fab: Content-based, collaborative recommendation. *Commun. ACM* 40, 3 (1997), 66–72.

[15] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*. 119–126.

[16] Senjuti Basu Roy, Sihem Amer-Yahia, Ashish Chawla, Gautam Das, and Cong Yu. 2010. Space efficiency in group recommendation. *VLDB J.* 19, 6 (2010), 877–900.

[17] Daniel J. Beal, Robin R. Cohen, Michael J. Burke, and Christy L. McLendon. 2003. Cohesion and performance in groups: A meta-analytic clarification of construct relations. *J. Appl. Psychol.* 88, 6 (2003), 989.

[18] Shlomo Berkovsky and Jill Freyne. 2010. Group-based recipe recommendations: Analysis of data aggregation strategies. In *Proceedings of the 4th ACM Conference on Recommender Systems*. ACM, 111–118.

[19] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Model. User-adapt. Interact.* 12, 4 (2002), 331–370.

[20] Robin Burke. 2007. Hybrid web recommender systems. In *The Adaptive Web*. Springer, 377–408.

[21] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive group recommendation. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 645–654.

[22] Da Cao, Xiangnan He, Lianhai Miao, Guangyi Xiao, Hao Chen, and Jiao Xu. 2019. Social-enhanced attentive group recommendation. *IEEE Trans. Knowl. Data Eng.* (2019).

[23] Jorge Castro, Jie Lu, Guangquan Zhang, Yucheng Dong, and Luis Martínez. 2017. Opinion dynamics-based group recommender systems. *IEEE Trans. Syst. Man Cybernet.: Syst.* 48, 12 (2017), 2394–2406.

[24] Edgar Ceh-Varela and Huiping Cao. [n.d.]. Github repository for Aggregation-based Group Recommender Systems for Ephemeral Groups. Retrieved from https://github.com/ecehvarela/grecsysbenchmark, year=2021.

[25] Edgar Ceh-Varela and Huiping Cao. 2019. Recommending Packages of Multi-Criteria Items to groups. In *Proceedings of the IEEE International Conference on Web Services (ICWS'19)*. IEEE, 273–282.

[26] Edgar Ceh-Varela, Huiping Cao, and Hady W. Lauw. 2021. Retrieved from https://computerscience.nmsu.edu/_files/documents/Technical_Report_Edgar_2021_0224.pdf.

[27] Amra Delic, Francesco Ricci, and Julia Neidhardt. 2019. Preference networks and non-linear preferences in group recommendations. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. ACM, 403–407.

[28] Li Duan, Tieliang Gao, Wei Ni, and Wei Wang. 2021. A hybrid intelligent service recommendation by latent semantics and explicit ratings. *Int. J. Intell. Syst.* 36, 12 (2021), 7867–7894.
[29] Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič. 2018. Evaluating group recommender systems. In *Group Recommender Systems*. Springer, 59–71.
[30] Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič. 2018. *Group Recommender Systems: An Introduction*. Springer.
[31] Guillermo Fernández, Waldemar López, Fernando Olivera, Bruno Rienzi, and Pablo Rodríguez-Bocca. 2014. Let's go to the cinema! A movie recommender system for ephemeral groups of users. In *Proceedings of the XL Latin American Computing Conference (CLEI'14)*. IEEE, 1–12.
[32] Noah E. Friedkin. 2004. Social cohesion. *Annu. Rev. Sociol.* 30 (2004), 409–425.
[33] Mike Gartrell, Xinyu Xing, Qin Lv, Aaron Beach, Richard Han, Shivakant Mishra, and Karim Seada. 2010. Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM International Conference on Supporting Group Work*. 97–106.
[34] Sarik Ghazarian and Mohammad Ali Nematbakhsh. 2015. Enhancing memory-based collaborative filtering for group recommender systems. *Expert Syst. Appl.* 42, 7 (2015), 3801–3812.
[35] Ram Deepak Gottapu and Lakshmi Venkata Sriram Monangi. 2017. Point-of-interest recommender system for social groups. *Proc. Comput. Sci.* 114 (2017), 159–164.
[36] Lei Guo, Hongzhi Yin, Tong Chen, Xiangliang Zhang, and Kai Zheng. 2021. Hierarchical hyperedge embedding-based representation learning for group recommendation. arXiv:2103.13506. Retrieved from https://arxiv.org/abs/2103.13506.
[37] Lei Guo, Hongzhi Yin, Qinyong Wang, Bin Cui, Zi Huang, and Lizhen Cui. 2020. Group recommendation with latent voting mechanism. In *Proceedings of the IEEE 36th International Conference on Data Engineering (ICDE'20)*. IEEE, 121–132.
[38] Xiaotian Han, Chuan Shi, Lei Zheng, Philip S. Yu, Jianxin Li, and Yuanfu Lu. 2018. Representation learning with depth and breadth for recommendation using multi-view data. In *Proceedings of the Asia-Pacific Web and Web-Age Information Management Joint International Conference on Web and Big Data (APWeb/WAIM'18)*. Springer, 181–188.
[39] F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2015), 1–19.
[40] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. 507–517.
[41] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. arXiv:1808.03912. Retrieved from https://arxiv.org/abs/1808.03912.
[42] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.
[43] Zhixiang He, Chi-Yin Chow, and Jia-Dong Zhang. 2020. GAME: Learning graphical and attentive multi-view embeddings for occasional group recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 649–658.
[44] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (2004), 5–53.
[45] Peng Huang, Nicholas H. Lurie, and Sabyasachi Mitra. 2009. Searching for experience on the web: An empirical examination of consumer behavior for search and experience goods. *J. Market.* 73, 2 (2009), 55–69.
[46] Won-Seok Hwang, Juan Parc, Sang-Wook Kim, Jongwuk Lee, and Dongwon Lee. 2016. "Told you i didn't like it" Exploiting uninteresting items for effective collaborative filtering. In *Proceedings of the IEEE 32nd International Conference on Data Engineering (ICDE'16)*. IEEE, 349–360.
[47] Heung-Nam Kim and Abdulmotaleb El Saddik. 2015. A stochastic approach to group recommendations in social media systems. *Inf. Syst.* 50 (2015), 76–93.
[48] Jaekyeong Kim, Ilyoung Choi, and Qinglong Li. 2021. Customer satisfaction of recommender system: Examining accuracy and diversity in several types of recommendation approaches. *Sustainability* 13, 11 (2021), 6165.
[49] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[50] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–A survey. *Knowl.-Bas. Syst.* 123 (2017), 154–162.
[51] Guohui Li, Qi Chen, Bolong Zheng, Hongzhi Yin, Quoc Viet Hung Nguyen, and Xiaofang Zhou. 2020. Group-based recurrent neural networks for POI recommendation. *ACM Trans. Data Sci.* 1, 1 (2020), 1–18.
[52] Xingjie Liu, Yuan Tian, Mao Ye, and Wang-Chien Lee. 2012. Exploring personal impact for group recommendation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. 674–683.

[53] Judith Masthoff. 2004. Group modeling: Selecting a sequence of television items to suit a group of viewers. In *Personalized Digital Television*. Springer, 93–141.
[54] Judith Masthoff. 2011. Group recommender systems: Combining individual models. In *Recommender Systems Handbook*. Springer, 677–702.
[55] Hanan Mengash and Alexander Brodsky. 2016. Tailoring group package recommendations to large heterogeneous groups based on multi-criteria optimization. In *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS'16)*. IEEE, 1537–1546.
[56] Diego Monti, Giuseppe Rizzo, and Maurizio Morisio. 2021. A systematic literature review of multicriteria recommender systems. *Artif. Intell. Rev.* 54 (2021), 427–468.
[57] Susan M. Mudambi and David Schuff. 2010. What makes a helpful review? A study of customer reviews on Amazon.com. *MIS Quart.* 34, 1 (2010), 185–200.
[58] Lihi Naamani-Dery, Meir Kalech, Lior Rokach, and Bracha Shapira. 2014. Preference elicitation for narrowing the recommended list for groups. In *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, 333–336.
[59] Le Nguyen Hoai Nam, Ho Thi Hoang Vy, Le Hoang My, Le Thi Tuyet Mai, Hong Tiet Gia, and Ho Le Thi Kim Nhung. 2019. An approach to improving group recommendation systems based on latent factor matrices. In *Proceedings of the 10th International Symposium on Information and Communication Technology*. 98–105.
[60] Reza Barzegar Nozari and Hamidreza Koohi. 2020. A novel group recommender system based on members' influence and leader impact. *Knowl.-Bas. Syst.* 205 (2020), 106296.
[61] Mark O'connor, Dan Cosley, Joseph A. Konstan, and John Riedl. 2001. PolyLens: A recommender system for groups of users. In *Proceedings of the European Conference on Computer-Supported Cooperative Work (ECSCW'01)*. Springer, 199–218.
[62] Zacharoula Papamitsiou and Anastasios A. Economides. 2018. Can't get more satisfaction?: Game-theoretic group-recommendation of educational resources. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. ACM, 409–416.
[63] Shameem A. Puthiya Parambath, Nishant Vijayakumar, and Sanjay Chawla. 2018. Saga: A submodular greedy algorithm for group recommendation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.
[64] Alexander Pelaez, Martin Y. Yu, and Karl R. Lang. 2013. Social buying: The effects of group size and communication on buyer performance. *Int. J. Electr. Commerc.* 18, 2 (2013), 127–157.
[65] Abinash Pujahari and Vineet Padmanabhan. 2015. Group recommender systems: Combining user-user and item-item collaborative filtering techniques. In *Proceedings of the International Conference on Information Technology (ICIT'15)*. IEEE, 148–152.
[66] Abinash Pujahari and Dilip Singh Sisodia. 2020. Aggregation of preference relations to enhance the ranking quality of collaborative filtering based group recommender system. *Expert Syst. Appl.* (2020), 113476.
[67] Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2016. Recommending packages to groups. In *Proceedings of the IEEE 16th International Conference on Data Mining (ICDM'16)*. IEEE, 449–458.
[68] Lara Quijano-Sanchez, Juan A. Recio-Garcia, Belen Diaz-Agudo, and Guillermo Jimenez-Diaz. 2013. Social factors in group recommender systems. *ACM Trans. Intell. Syst. Technol.* 4, 1 (2013), 1–30.
[69] Elisa Quintarelli, Emanuele Rabosio, and Letizia Tanca. 2016. Recommending new items to ephemeral groups using contextual user influence. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 285–292.
[70] Xun Ran, Yong Wang, Leo Yu Zhang, and Jun Ma. 2022. A differentially private nonnegative matrix factorization for recommender system. *Inf. Sci.* (2022).
[71] Alan Said and Alejandro Bellogín. 2014. Comparative recommender system evaluation: Benchmarking recommendation frameworks. In *Proceedings of the 8th ACM Conference on Recommender Systems*. 129–136.
[72] Maria Salamó, Kevin McCarthy, and Barry Smyth. 2012. Generating recommendations for consensus negotiation in group personalization services. *Pers. Ubiq. Comput.* 16, 5 (2012), 597–610.
[73] Aravind Sankar, Yanhong Wu, Yuhang Wu, Wei Zhang, Hao Yang, and Hari Sundaram. 2020. GroupIM: A mutual information maximization framework for neural group recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1279–1288.
[74] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms. *WWW* 1 (2001), 285–295.
[75] Young-Duk Seo, Young-Gab Kim, Euijong Lee, Kwang-Soo Seol, and Doo-Kwon Baik. 2018. An enhanced aggregation method considering deviations for a group recommendation. *Expert Syst. Appl.* 93 (2018), 299–312.
[76] Jing Shi, Bin Wu, and Xiuqin Lin. 2015. A latent group model for group recommendation. In *Proceedings of the IEEE International Conference on Mobile Services*. IEEE, 233–238.
[77] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. 2019. How good your recommender system is? A survey on evaluations in recommendation. *Int. J. Mach. Learn. Cybernet.* 10, 5 (2019), 813–831.
[78] Philip E. Slater. 1958. Contrasting correlates of group size. *Sociometry* 21, 2 (1958), 129–139.

[79] Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management*. 623–632.
[80] Maria Stratigi, Jyrki Nummenmaa, Evaggelia Pitoura, and Kostas Stefanidis. 2020. Fair sequential group recommendations. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. 1443–1452.
[81] Lifeng Sun, Xiaoyan Wang, Zhi Wang, Hong Zhao, and Wenwu Zhu. 2016. Social-aware video recommendation for online social groups. *IEEE Trans. Multimedia* 19, 3 (2016), 609–618.
[82] Poonam B. Thorat, R. M. Goudar, and Sunita Barve. 2015. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *Int. J. Comput. Appl.* 110, 4 (2015), 31–36.
[83] Joseph K. Torgesen. 2006. Intensive reading interventions for struggling readers in early elementary school: A principal's guide. Center on Instruction.
[84] Ruth Towse and Trilce Navarrete Hernández. 2020. *Handbook of Cultural Economics*. Edward Elgar Publishing.
[85] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 515–524.
[86] Ximeng Wang, Yun Liu, Jie Lu, Fei Xiong, and Guangquan Zhang. 2019. TruGRC: Trust-aware group recommendation with virtual coordinators. *Fut. Gener. Comput. Syst.* 94 (2019), 224–236.
[87] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Wei Chen, and Tie-Yan Liu. 2013. A theoretical analysis of NDCG ranking measures. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT'13)*, Vol. 8. 6.
[88] Susan A. Wheelan. 2009. Group size, group development, and group productivity. *Small Group Res.* 40, 2 (2009), 247–262.
[89] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Npa: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Dtentiva Mining*. ACM, 2576–2584.
[90] Zhengzheng Xian, Qiliang Li, Gai Li, and Lei Li. 2017. New collaborative filtering algorithms based on SVD. *Math. Probl. Eng.* (2017).
[91] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. 2019. Social influence-based group representation learning for group recommendation. In *Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE'19)*. IEEE, 566–577.
[92] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, and Xiaofang Zhou. 2020. Overcoming data sparsity in group recommendation. *IEEE Trans. Knowl. Data Eng.*
[93] Quan Yuan, Gao Cong, and Chin-Yew Lin. 2014. COM: A generative model for group recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 163–172.
[94] Alfredo Zapata, Víctor H. Menéndez, Manuel E. Prieto, and Cristóbal Romero. 2015. Evaluation and selection of group recommendation strategies for collaborative searching of learning objects. *Int. J. Hum.-Comput. Stud.* 76 (2015), 22–39.
[95] Jia-Dong Zhang and Chi-Yin Chow. 2018. SEMA: Deeply learning semantic meanings and temporal dynamics for recommendations. *IEEE Access* 6 (2018), 54106–54116.
[96] Qiliang Zhu and Lei Wang. 2020. Context-aware restaurant recommendation for group of people. In *Proceedings of the IEEE World Congress on Services (SERVICES'20)*. IEEE, 51–54.

Received October 2021; revised March 2022; accepted April 2022