Group Recommendations by Learning Rating Behavior

Dimitris Sacharidis
TU Wien
E-Commerce Group
Austria
dimitris@ec.tuwien.ac.at

ABSTRACT

In many domains, it is often required to provide recommendations for groups, instead of individual users. Existing approaches try to compensate for the lack of group profiles, by either merging individual profiles, or treating users separately and then fusing the recommendations. Both paradigms thus fail to account for the different roles and behaviors people assume when making group decisions. In this work, we propose two novel group recommendation models that explicitly try to model the behavior of group members and distinguish it from that when they act alone. A detailed evaluation has shown that our models consistently provide significantly better recommendations. In addition, useful conclusions are drawn regarding the favorable settings of existing techniques.

KEYWORDS

Collaborative Filtering; Group Recommendation; Latent Factor Model

ACM Reference format:
Dimitris Sacharidis. 2017. Group Recommendations by Learning Rating Behavior. In *Proceedings of UMAP '17, Bratislava, Slovakia, July 09-12, 2017,* 9 pages.
DOI: http://dx.doi.org/10.1145/3079628.3079691

1 INTRODUCTION

Recommender systems are nowadays ubiquitous, providing recommendations in diverse domains, e.g., for movies/tv programs (Netflix), e-commerce (Amazon), music (Spotify), apps (Apple App Store and Google Play), books (Goodreads). Usually, the underlying mechanism for providing recommendations, follows the principles of collaborative filtering (CF), where the idea is to leverage the observed interests and ratings from other users [22]. More recently, and following their success at the Netflix prize, *latent factor models* have become the standard in materializing the CF idea [10, 11].

While traditional research on recommender systems has almost exclusively focused on providing recommendations to *single* users, there exist many cases, where the system needs to suggest items to *groups* of users [1, 4, 16–18, 25]. As examples, consider a group of friends seeking to go together on a vacation, or a family that decides on a movie to watch at home. Existing methods for group recommendations basically follow one of two paradigms. The first, hereafter termed PROF-AGG, is to explicitly construct a group profile by combining (aggregating) the profiles of individual members. In this way, the group can be treated as a *pseudo user*, and thus standard techniques can be employed to provide recommendations for the group. The second paradigm, termed REC-AGG, is to first compute recommendations for each member separately, and then employ an aggregation strategy across them to compile the group recommendations. Inspired by social choice theory, numerous aggregation strategies for profiles and recommendations exist.

These two paradigms share some drawbacks. First, they assume certain decision dynamics within an group, i.e., the aggregation strategy, and often fix on this. Second, they treat group members the same as individual users, mostly assuming that the behavior and preferences of users in groups is identical to that when they decide alone. Of course, there are some notable exceptions that avoid these drawbacks. For instance the hybrid switching strategy of [3] (albeit between group, general, and individual recommendations) for the former, INTRIGUE [1] where not all members are treated equally, [21] that includes personality and social trust, and [7] that considers relationship strength for the latter. However these works rely on additional information about group members, which one cannot assume in a pure CF setting.

For groups that are relatively long-standing, it is reasonable to expect that sufficient information has been collected in order to build a group profile, eliminating the need for artificial profile aggregation. However, we note that in this case the recommender may suffer from cold-start problems. For instance, the system would not be able to assess *cold items* for which no or very few ratings by any group is available, even though this item may have been rated by individual users. Similarly, such a system cannot provide recommendations for *cold groups*, with no or little group profile, even though members of the groups may have individual profiles. An approach to counter these cold-start problems would be to employ PROF-AGG, which however defeats the purpose at it inherits its shortcomings.

In this work, we propose methods that address the aforementioned issues. We assume a pure collaborative filtering setting, where only a history of user-item ratings along with a few group-item ratings are available. This is a reasonable assumption, as past work has considered such sparse group profiles [8, 12], and a relevant challenge was set up [19]. The problem we address is how to best exploit the group and user profiles for group recommendations. We base our approach on the understanding that people assume different roles (e.g., leaders or followers) when in groups, or even across groups (e.g., in work and in family), and thus may exhibit substantially different behavior compared to them acting individually. Our proposed methods

UMAP '17, Bratislava, Slovakia

978-1-4503-4635-1/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3079628.3079691

attempt to explicitly *learn the discrepancies between individual and group rating behavior*.

Our first model, termed RESIDUAL, presupposes that the group rating differs from the average member rating by a sum of *residuals* for each group member. For a particular user and a particular group, her residual rating captures the difference between her individual and group ratings averaged over all items in the group profile. As only very few of these residuals can actually be computed, due to the sparsity in the group profiles, we employ matrix factorization to predict the missing residuals. RESIDUAL can thus account for different rating behavior of users within groups and across groups. Our second model, termed TRIAD, presupposes that group ratings are computed as a weighted average of member ratings. Therefore, the weight of a particular user captures her behavior change as a group member. TRIAD learns the user behavior weights together with conventional latent factors by jointly examining user and group ratings. While both models assume that group ratings are generated by a linear combination of individual ratings, our evaluation shows that they perform well even when this may not be the case, e.g., in least-misery or dictatorship situations.

Evaluating group recommendations is a particularly complex task. Even when ground truth data is available, a rather rare sighting [5, 12, 24], measuring the satisfaction of user in group decisions remains an open research topic [15]. To handle the absence of real data, researchers typically resort to one of two approaches. In the first, the group rating of an item is synthesized to be the average (or some other aggregate) of individual ratings. Then, one can use standard evaluation metrics to quantify how far the predicted group ratings is from the synthesized. In the second approach, the predicted group recommendations are evaluated for each group member individually, and then averaged [2, 3]. As both approaches compute some average satisfaction, they tend to favor average aggregation strategies as remarked in [14].

More generally, presuming a particular group satisfaction metric (e.g., the average member satisfaction) naturally introduces bias. To combat this, we make a simple but significant contribution towards an unbiased evaluation of group recommenders under synthetic datasets. We construct multiple sets of synthetic group ratings assuming different group satisfaction criteria, and measure the performance of a group recommender at each. Then, a method consistently outperforming others under multiple criteria constitutes a stronger and less biased indication of its effectiveness.

We perform a detailed evaluation on real and synthetically generated group ratings under a broad range of group aggregation strategies. We compare existing methods and identify their favorable settings. More importantly, we evaluate our two proposed models and find that they are significantly more accurate than the standard methods in all settings and under multiple evaluation criteria.

The remainder of this paper is structured as follows. Section 2 defines the problem and establishes the necessary background describing existing group recommendation methods. Section 3 presents our two proposed models for group recommendations. Then, Section 4 presents a thorough experimental evaluation of existing work and ours. Finally, Section 5 concludes the paper.

2 PROBLEM DEFINITION AND BACKGROUND

In Section 2.1, we first formally define the group recommendation problem. Then, in Section 2.2 we briefly overview a simple latent factor model. In Section 2.3, we categorize existing work on group recommendations.

2.1 Problem Definition

We consider a set of users $\mathcal{U} = \{u_i\}$, a set of items $\mathcal{V} = \{v_j\}$, and a set of groups $\mathcal{G} = \{g_k\} \subseteq 2^{\mathcal{U}}$; we use the subscripts i, j, k to refer to an individual user, item, or group, respectively. Further, we assume we have a set $\mathcal{R}^{\mathcal{U}}$ of user-item ratings, and a set $\mathcal{R}^{\mathcal{G}}$ of group-item ratings. Then, the problem of recommending items to groups can be abstractly stated as follows.

Problem 1. [Collaborative Filtering Group Recommendation] Given $\mathcal{U}$, $\mathcal{V}$, $\mathcal{G}$ and ratings $\mathcal{R}^{\mathcal{U}}$, $\mathcal{R}^{\mathcal{G}}$, predict for each group $g_k \in \mathcal{G}$ the rating of each not previously consumed item.

Throughout this paper, we follow the notational convention that bold small letters, e.g., $\mathbf{x}$, indicate column vectors, and bold capital letters, e.g., $\mathbf{A}$, denote matrices.

2.2 Matrix Factorization for Recommendations

A family of very popular techniques for providing item recommendations is the latent factor models, also known as *matrix factorization* [10]. The basic idea is to view the user-item ratings as a sparse matrix, for which we wish to predict the values of its empty cells. This is achieved by computing a low-rank approximation of the rankings matrix.

Specifically, we assume that each user u_i is associated with an f-dimensional factor (column) vector $\mathbf{u}_i$, and similarly each item v_j with an f-dimensional factor vector $\mathbf{v}_j$. Then, the predicted rating of item v_j by user u_i is computed as the inner product of the corresponding factor vectors:

$$\hat{r}_{ij} = \mathbf{u}_i^T \mathbf{v}_j. \tag{1}$$

Under this model, the objective is then to compute the factor vectors of each user and item so that they provide accurate estimations of the known ratings without overfitting. Note that there are many ways to formulate this goal; here we consider the simplest approach which minimizes the regularized squared error on the set of ratings [10]:

$$\sum_{(u_i, v_j) \in \mathcal{R}^{\mathcal{U}}} (r_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 - \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2),$$

where $\mathbf{U}$, $\mathbf{V}$ denote the $f \times |\mathcal{U}|$ user matrix and the $f \times |\mathcal{V}|$ item matrix, respectively, consisting of all user and item factor vectors, λ is a parameter controlling the extent of regularization, and $\|\mathbf{A}\|_F$ denotes the Frobenius norm of matrix $\mathbf{A}$ used for regularization.

To minimize the objective function and determine the factor vectors, one can apply standard techniques, e.g., Alternating Least Squares (ALS), or Stochastic Gradient Descent (SGD).

2.3 Group Recommender Systems

Literature on group recommenders is rich; we refer the reader to [9, 14] for a systematic treatment of this research area. In this work, we consider collaborative filtering techniques, focus on the task of

recommending a single item to the group, and optimize primarily for the prediction error, i.e., the difference between the predicted and the actual group rating.

In the absence of group profiles, the recommender system needs to compensate. There are two basic paradigms in which an existing recommender for individual users can be extended to provide group recommendations. In PROF-AGG, also referred to as aggregated model [3] or group model [13], a group profile is created by aggregating the profiles of group members. In REC-AGG, recommendations for group members are compiled independently, and are then fused to create group recommendations. Essentially, in the CF case, both paradigms perform an aggregation of either actual or predicted ratings.

There are numerous aggregation strategies that one can employ in either paradigm. These are mostly inspired by social choice theory ideas; see [13] for an overview, and a study on how people select recommendations for a group so as to balance the preferences of the group members. Popular strategies include taking the average, the minimum a.k.a. least misery principle of not strongly displeasing any member, the maximum for satisfying the maximum pleasure among members, the product. For reference we mention the following: MusicFX [16] implements a least misery criterion in group modeling (PROF-min); POLYLENS [18] aggregates recommendations assuming least misery (REC-min); INTRIGUE [1] is an interesting hybrid that identifies sub-groups among groups (e.g., children, or disabled persons), creates a model for each sub-group (PROF-AGG), and then fuses the sub-group recommendations under a weighted scheme (REC-avg); Yu's TV recommender [25] constructs group profiles so as to minimize distance among individual profiles (PROF-AGG); the content-based TV recommender in [24] investigates the optimal aggregation strategy for group modeling (PROF-AGG).

A significant line of work concerns the evaluation of group recommenders. The seminal work of [15] studies what factors influence group satisfaction and how it differs from individual satisfaction. A comparison of PROF-AGG strategies can be found at [23]. Various CF-based rank aggregation techniques (e.g., [6]) for REC-AGG are examined in [2]. A comparison of both PROF-AGG and REC-AGG CF-based techniques in [3] concludes that group profile modeling is better than recommendation aggregation.

Other approaches for group recommendations have also appeared. Most notably, the information matching approach of [8], denoted as INF-MATCH, predicts the relevance (instead of the rating) of items to groups. Each user is assumed to give ratings according to a 2-Poisson mixture model, where one component describes the ratings for relevant items (those with rating above some threshold), and the other those for irrelevant or unrated items. Similarly, each item receives ratings according to another 2-Poisson mixture model. INF-MATCH predicts the relevance probability of an item to a user by taking into account the mixture models of all items and users. Then, to predict the relevance probability of an item to a group, INF-MATCH follows the Least Misery strategy, assigning to the group the minimum relevance probability among its members.

3 OUR GROUP RECOMMENDER MODELS

In this section, we present our contributions to the group recommendation problem. In Section 3.1, we describe RESIDUAL that tries to estimate the difference in the behavior of a user in a group and alone. In Section 3.2, we present TRIAD that learns the strength with which a user enforces her preferences in a group.

3.1 The Residual Model

The residual model, denoted as RESIDUAL, predicts the rating of an item v_j by group g_k as:

$$\hat{r}_{kj} = \frac{1}{|g_k|} \sum_{u_i \in g_k} (\hat{r}_{ij} + \hat{\xi}_{ik}), \tag{2}$$

where $\hat{r}_{ij}$ is the predicted rating (according to some model) of item v_j by user u_i, and $\hat{\xi}_{ik}$ is the *predicted residual rating* of user u_i when in group g_k. The intuition, here is that the actual group rating r_{kj} differs from the average user rating by some (unknown, but estimated) group-specific user residual ξ_{ik}. Note that when $\hat{\xi} = 0$ for all users and groups, the RESIDUAL model essentially becomes the REC-avg technique described in Section 2.3.

In the following, we discuss how we compute the predicted residual ratings. Consider a user u_i being a member of group g_k, and let $\mathcal{V}_k$ denote the set of items group g_k has rated. Then, given ratings $\mathcal{R}^{\mathcal{G}}$, and a model for predicting user-item ratings, we define the *residual rating* ξ_{ik} of user u_i in group g_k as:

$$\xi_{ik} = \frac{1}{|\mathcal{V}_k|} \sum_{v_j \in \mathcal{V}_k} (r_{kj} - \hat{r}_{ij}). \tag{3}$$

Now, consider the sparse $|\mathcal{U}| \times |\mathcal{G}|$ matrix Ξ, which has value ξ_{ik} computed as above when user $u_i \in g_k$, value $\xi_{ik} = 0$ when $u_i \notin g_k$, and unknown value elsewhere. We factorize the Ξ matrix into a $f_{\xi} \times |\mathcal{U}|$ matrix $\mathbf{P}$ and a $f_{\xi} \times |\mathcal{G}|$ matrix $\mathbf{Q}$, using a technique similar to that in Section 2.2, to obtain a rank f_{ξ} approximation. Then, the predicted residual rating of user u_i when in group g_k is:

$$\hat{\xi}_{ik} = \mathbf{p}_i^T \mathbf{q}_k,$$

where $\mathbf{p}_i$ is the factor vector in matrix $\mathbf{P}$ corresponding to user u_i, and $\mathbf{q}_k$ is the factor vector in matrix $\mathbf{Q}$ corresponding to group g_k.

To better understand the reasoning behind the RESIDUAL model, consider the following. Assume that we have a complete set of ratings for group g_k, i.e., all items have a group rating and thus $\mathcal{V}_k = \mathcal{V}$. Then, for each member u_i of g_k, its predicted residual rating is equal to its residual rating $\frac{1}{|\mathcal{V}|} \sum_{v_j \in \mathcal{V}} (r_{kj} - \hat{r}_{ij})$. Replacing this value into Equation 2 and expanding we obtain:

$$\hat{r}_{kj} = \frac{1}{|g_k|} \sum_{u_i \in g_k} \hat{r}_{ij} + \frac{1}{|\mathcal{V}|} \sum_{v_j \in \mathcal{V}} r_{kj} - \frac{1}{|g_k||\mathcal{V}|} \sum_{u_i \in g_k} \sum_{v_j \in \mathcal{V}} \hat{r}_{ij}.$$

Then computing the mean of $\hat{r}_{kj}$ over all items $v_j \in V$, we derive:

$$E_{v_j \in V}[\hat{r}_{kj}] = \frac{1}{|\mathcal{V}|} \sum_{v_j \in \mathcal{V}} r_{kj},$$

where $E_{v_j \in V}[]$ denotes expectation over items v_j taken uniformly at random from V. In other words, the mean predicted rating of group g_k is equal to the average actual rating that group g_k gives to items.

Learning. In our implementation of the RESIDUAL model, we use matrix factorization as the underlying model for predicting user-item ratings, which requires matrices $\mathbf{U}$, $\mathbf{V}$ as its parameters, as defined in Section 2.2. In addition, RESIDUAL requires two additional matrices $\mathbf{P}$, $\mathbf{Q}$ factorizing matrix Ξ of residual ratings as described.

First, we learn parameters $\mathbf{U}$, $\mathbf{V}$ from the set $\mathcal{R}^{\mathcal{U}}$ of user-item ratings using SGD, similar to [10]; details are omitted. Next, we compute the set of non-empty entries of matrix Ξ according to Equation 3. Finally, we learn $\mathbf{P}$, $\mathbf{Q}$ using SGD on the aforementioned set of Ξ entries; details are omitted.

3.2 The Triad Model

The triad model, denoted as TRIAD, predicts the rating of an item v_j by group g_k as:

$$\hat{r}_{kj} = \frac{1}{|g_k|} \sum_{u_i \in g_k} b_i \cdot \hat{r}_{ij}, \tag{4}$$

where $\hat{r}_{ij}$ is the predicted rating of item v_j by user u_i, and b_i is the *group behavior* of user u_i. The intuition here is that each user has a global (unknown) behavior when she becomes a member of a group. Similar to RESIDUAL, setting $b_i = 1$ for all users, makes the TRIAD model identical to the REC-avg technique described in Section 2.3.

TRIAD uses matrix factorization to predict user-item ratings (Section 2.2). Therefore, it employs matrices $\mathbf{U}$, $\mathbf{V}$ and Equation 1 to predict $\hat{r}_{ij}$. In addition, the TRIAD model uses a third parameter — hence its name, the $|\mathcal{U}|$-dimensional group behavior vector $\mathbf{b}$, containing the group behavior of each user.

To rewrite Equation 4 using matrices, we introduce some additional notation. For any group g_k, let $\mathbf{g}_k$ denote its $|\mathcal{U}|$-dimensional *user membership vector*, where its i-th coordinate has value $\frac{1}{|g_k|}$ if user u_i is a member of g_k, and zero otherwise. Moreover, let symbol $\circ$ denote the element-wise (Hadamard) product for vectors: $(\mathbf{x} \circ \mathbf{y})_{[i]} = \mathbf{x}_{[i]}\mathbf{y}_{[i]}$. Then, Equation 4 is equivalent to:

$$\hat{r}_{kj} = (\mathbf{g}_k \circ \mathbf{b})^T \mathbf{U}^T \mathbf{v}_j. \tag{5}$$

Observe that the $(\mathbf{g}_k \circ \mathbf{b})^T$ matrix essentially plays the role of the $\frac{1}{|g_k|} \sum_{u_i \in g_k} b_i$ part in Equation 4.

Learning. In what follows we describe how to learn the TRIAD parameters. Note that contrary to RESIDUAL, all model parameters are learned together in one phase, using the user-item *and* the group-item ratings.

We make the assumption that each observed user rating r_{ij} (resp. group rating r_{kj}) follows a Gaussian distribution with mean $\hat{r}_{ij}$ (resp. $\hat{r}_{kj}$) and variance σ^2.

Therefore, given rankings $\mathcal{R}^{\mathcal{U}}$, $\mathcal{R}^{\mathcal{G}}$, the likelihood of the TRIAD model is:

$$p(\mathcal{R}^{\mathcal{U}}, \mathcal{R}^{\mathcal{G}} | \mathbf{U}, \mathbf{V}, \mathbf{b}) = \prod_{(i,j) \in \mathcal{R}^{\mathcal{U}}} \mathcal{N}(r_{ij}; \hat{r}_{ij}, \sigma^2) \prod_{(k,j) \in \mathcal{R}^{\mathcal{G}}} \mathcal{N}(r_{kj}; \hat{r}_{kj}, \sigma^2)$$

where $\mathcal{N}(x; \mu, \sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2.

We next assign spherical Gaussian priors [20] on the model parameters:

$$p(\mathbf{U}|\sigma_{\mathbf{U}}^2) = \prod_{u_i} \mathcal{N}(\mathbf{u}_i; 0, \sigma_{\mathbf{U}}^2 \mathbf{I})$$
$$p(\mathbf{V}|\sigma_{\mathbf{V}}^2) = \prod_{v_j} \mathcal{N}(\mathbf{v}_j; 0, \sigma_{\mathbf{V}}^2 \mathbf{I})$$
$$p(\mathbf{b}|\sigma_{\mathbf{b}}^2) = \prod_{u_i} \mathcal{N}(b_i; 0.5, \sigma_{\mathbf{b}}^2)$$

We seek to maximize the posterior:

$$p(\mathbf{U}, \mathbf{V}, \mathbf{b} | \mathcal{R}^{\mathcal{U}}, \mathcal{R}^{\mathcal{G}}, \sigma_\theta) \propto p(\mathcal{R}^{\mathcal{U}}, \mathcal{R}^{\mathcal{G}} | \mathbf{U}, \mathbf{V}, \mathbf{b}) p(\mathbf{U}|\sigma_{\mathbf{U}}^2) p(\mathbf{V}|\sigma_{\mathbf{V}}^2) p(\mathbf{b}|\sigma_{\mathbf{b}}^2)$$

where $\sigma_\theta^2 = \{\sigma^2, \sigma_{\mathbf{U}}^2, \sigma_{\mathbf{T}}^2, \sigma_{\mathbf{b}}^2\}$, or minimize correspondingly the negative log posterior, which is equivalent (after eliminating terms depending on σ_θ^2 and constants) to minimizing the following regularized error function:

$$E = \sum_{(i,j) \in \mathcal{R}^{\mathcal{U}}} \left(r_{ij} - \mathbf{u}_i^T \mathbf{v}_j \right)^2 + \sum_{(k,j) \in \mathcal{R}^{\mathcal{G}}} \left(r_{kj} - (\mathbf{g}_k \circ \mathbf{b})^T \mathbf{U}^T \mathbf{v}_j \right)^2 + \\ + \lambda_{\mathbf{U}} \|\mathbf{U}\|_F^2 + \lambda_{\mathbf{V}} \|\mathbf{V}\|_F^2 + \lambda_{\mathbf{b}} \|\mathbf{b}'\|^2, \tag{6}$$

where $\lambda_{\mathbf{U}} = \sigma_{\mathbf{U}}^2/\sigma^2$, $\lambda_{\mathbf{V}} = \sigma_{\mathbf{V}}^2/\sigma^2$, $\lambda_{\mathbf{b}} = \sigma_{\mathbf{b}}^2/\sigma^2$, and $\mathbf{b}' = \mathbf{b} - 0.5\mathbf{I}$.

To learn the parameters $\mathbf{U}$, $\mathbf{V}$, $\mathbf{b}$ of TRIAD from the sets of ratings $\mathcal{R}^{\mathcal{U}}$, $\mathcal{R}^{\mathcal{G}}$, we perform SGD as follows. We rewrite Equation 6 as:

$$E = \sum_{(i,j) \in \mathcal{R}^{\mathcal{U}}} E_{\mathcal{R}^{\mathcal{U}}} + \sum_{(k,j) \in \mathcal{R}^{\mathcal{G}}} E_{\mathcal{R}^{\mathcal{G}}},$$

where:

$$E_{\mathcal{R}^{\mathcal{U}}} = \left(r_{ij} - \mathbf{u}_i^T \mathbf{v}_j \right)^2 + \lambda'_{\mathbf{U}} \|\mathbf{U}\|_F^2 + \lambda'_{\mathbf{V}} \|\mathbf{V}\|_F^2$$
$$E_{\mathcal{R}^{\mathcal{G}}} = \left(r_{kj} - (\mathbf{g}_k \circ \mathbf{b})^T \mathbf{U}^T \mathbf{v}_j \right)^2 + \lambda'_{\mathbf{U}} \|\mathbf{U}\|_F^2 + \lambda'_{\mathbf{V}} \|\mathbf{V}\|_F^2 + \lambda'_{\mathbf{b}} \|\mathbf{b}'\|^2,$$

and $\lambda'_{\mathbf{U}} = \lambda_{\mathbf{U}}/(|\mathcal{R}^{\mathcal{U}}| + |\mathcal{R}^{\mathcal{G}}|)$, $\lambda'_{\mathbf{V}} = \lambda_{\mathbf{V}}/(|\mathcal{R}^{\mathcal{U}}| + |\mathcal{R}^{\mathcal{G}}|)$, $\lambda'_{\mathbf{b}} = \lambda_{\mathbf{b}}/|\mathcal{R}^{\mathcal{G}}|$.

Algorithm 1 presents the learning algorithm for TRIAD. Initially random values are chosen for the TRIAD parameters (Line 1). Then, the outer loop (Lines 2–15) is executed until convergence (or a maximum number of iterations is reached). In each iteration of the outer loop, all ratings from $\mathcal{R}^{\mathcal{U}}$ and $\mathcal{R}^{\mathcal{G}}$ are considered in the inner loop (Lines 3–14). On the other hand, in each iteration of the inner loop, a single rating is considered. This rating is chosen to be drawn from a set of ratings with probability proportional to the set's size. This is accomplished by flipping a coin with probability $|\mathcal{R}^{\mathcal{U}}|/(|\mathcal{R}^{\mathcal{U}}| + |\mathcal{R}^{\mathcal{G}}|)$ (random variable X sampled at Line 4). Once the set of ratings to draw from has been established, a rating is selected uniformly at random (Line 6 or 10). Then, the parameter values are updated in a gradient descent manner with learning rate η (Lines 7–9 or 11–13). An important thing to note is that a user rating r_{ij} updates vectors $\mathbf{u}_i$ and $\mathbf{v}_j$, whereas a group rating r_{kj} updates the user vectors of all g_k members (hence the matrix $\mathbf{U}$) and vectors $\mathbf{v}_j$ and $\mathbf{b}$.

To conclude the description of the learning algorithm, we need to compute the partial derivatives of $E_{\mathcal{R}^{\mathcal{U}}}$ and $E_{\mathcal{R}^{\mathcal{G}}}$ with respect to TRIAD parameters. Define the prediction error for user rating r_{ij} as $e_{ij} = r_{ij} - \mathbf{u}_i^T \mathbf{v}_j$. Then, the non-zero partial derivatives of $E_{\mathcal{R}^{\mathcal{U}}}$ are those with respect to all elements of vectors $\mathbf{u}_i$ and $\mathbf{v}_j$:

$$\frac{\partial E_{\mathcal{R}^{\mathcal{U}}}}{\partial \mathbf{u}_i} = -2e_{ij}\mathbf{v}_j + 2\lambda'_{\mathbf{U}}\mathbf{u}_i$$
$$\frac{\partial E_{\mathcal{R}^{\mathcal{U}}}}{\partial \mathbf{v}_j} = -2e_{ij}\mathbf{u}_i + 2\lambda'_{\mathbf{V}}\mathbf{v}_j.$$

Note that these derivatives are essentially identical to those used in the SGD for learning the standard matrix factorization model described in Section 2.2 (after setting $\lambda'_{\mathbf{U}} = \lambda'_{\mathbf{V}} = \lambda$).

Similarly, define the prediction error for group rating r_{kj} as $e_{kj} = r_{kj} - (\mathbf{g}_k \circ \mathbf{b})^T \mathbf{U}^T \mathbf{v}_j$. Then, the non-zero partial derivatives of $E_{\mathcal{R}^{\mathcal{G}}}$

Algorithm 1: TRIAD-Learn

Input: $\mathcal{R}^{\mathcal{U}}$, $\mathcal{R}^{\mathcal{G}}$, $\lambda'_{\mathbf{U}}$, $\lambda'_{\mathbf{V}}$, $\lambda'_{\mathbf{b}}$
Output: U, V, b
Variables: X Bernoulli random variable with probability $|\mathcal{R}^{\mathcal{U}}|/(|\mathcal{R}^{\mathcal{U}}| + |\mathcal{R}^{\mathcal{G}}|)$
Initialize U, V, b at random according to their priors
repeat
  repeat
    $x \leftarrow$ sample of Bernoulli random variable X
    if $x = 1$ **then**
      Draw pair (i, j) from $\mathcal{R}^{\mathcal{U}}$
      $\mathbf{u}_i \leftarrow \mathbf{u}_i - \eta \frac{\partial E_{\mathcal{R}^{\mathcal{U}}}}{\partial \mathbf{u}_i}$
      $\mathbf{v}_j \leftarrow \mathbf{v}_j - \eta \frac{\partial E_{\mathcal{R}^{\mathcal{U}}}}{\partial \mathbf{v}_j}$
    else
      Draw pair (k, j) from $\mathcal{R}^{\mathcal{G}}$
      $\mathbf{U} \leftarrow \mathbf{U} - \eta \frac{\partial E_{\mathcal{R}^{\mathcal{G}}}}{\partial \mathbf{U}}$
      $\mathbf{v}_j \leftarrow \mathbf{v}_j - \eta \frac{\partial E_{\mathcal{R}^{\mathcal{G}}}}{\partial \mathbf{v}_j}$
      $\mathbf{b} \leftarrow \mathbf{b} - \eta \frac{\partial E_{\mathcal{R}^{\mathcal{G}}}}{\partial \mathbf{b}}$
  until all ratings from $\mathcal{R}^{\mathcal{U}}$, $\mathcal{R}^{\mathcal{G}}$ are drawn
until convergence

are those with respect to elements of matrix $\mathbf{U}$ and vectors $\mathbf{v}_j$, $\mathbf{b}$:

$$\frac{\partial E_{\mathcal{R}^{\mathcal{U}}}}{\partial \mathbf{U}} = -2e_{kj}\mathbf{v}_j(\mathbf{g}_k \circ \mathbf{b})^T + 2\lambda'_{\mathbf{U}}\mathbf{U}$$
$$\frac{\partial E_{\mathcal{R}^{\mathcal{G}}}}{\partial \mathbf{v}_j} = -2e_{kj}(\mathbf{g}_k \circ \mathbf{b})^T\mathbf{U}^T + 2\lambda'_{\mathbf{V}}\mathbf{v}_j$$
$$\frac{\partial E_{\mathcal{R}^{\mathcal{G}}}}{\partial \mathbf{b}} = -2e_{kj}\mathbf{g}_k \circ \mathbf{U}^T\mathbf{v}_j + 2\lambda'_{\mathbf{b}}\mathbf{b}.$$

4 EXPERIMENTAL EVALUATION

In Section 4.1 we detail our experimental setting, describing the datasets and the evaluation metrics used. Then in Section 4.2 we present the results of our study.

4.1 Experimental Settings

4.1.1 Real Dataset. To evaluate our methods on a realistic setting, we use the data from the observational study of [5], henceforth denoted as REAL. Students from four universities were arranged into groups of 2–4 members. Each member was asked to individually rate on a 5-point scale the attractiveness of 11 popular European capitals as a touristic destination. Then, the groups convened and jointly agreed on their top-2 preferred destinations. Overall, there were 200 users partitioned across 60 groups.

For our purposes, we convert the ranking of destinations within each group into group rating scores using logarithmic discounting (e.g., as in the NDCG metric). Accordingly, the top ranking object receives the maximum score, while object at rank r receives the maximum score divided by $\log(1 + r)$. As only three ranks exist in our dataset, the top destination was rated with 5, the second with 3.15, and all the rest with 2.5. The resulting group ratings are split into training and test sets with a fixed ratio of 4:1.

4.1.2 Synthetic Datasets. We construct synthetic groups and group ratings based on the popular MovieLens 1M dataset[1]. It consists of $|\mathcal{U}| = 6,040$ users, $|\mathcal{V}| = 3,952$ items (movies), and $|\mathcal{R}^{\mathcal{U}}| = 1,000,209$ user-item ratings on an integer scale of 1 to 5. We note that this dataset contains no groups or group-item ratings.

[1] http://grouplens.org/datasets/movielens/

Table 1: Synthetic Group Ratings Parameters

Parameter	Symbol	Values	Default
group size	$\|g_k\|$	2 – 8	3
ratings per group	$\|\mathcal{V}_k\|$	50 – 200	100
training ratings to total	$\|\mathcal{V}_k^t\|/\|\mathcal{V}_k\|$	30% – 90%	80
relevance threshold	ρ	3	3

We synthetically construct groups, assigning users to groups uniformly at random. In each setting, all groups have the same number of members, denoted as $|g_k|$; we vary $|g_k|$ from 2 up to 8 users. In all settings, we keep the number of groups fixed to $|\mathcal{G}| = 50$, and the number of distinct users across all groups to 100.

Each group gives ratings to the same number of items $|\mathcal{V}_k|$, which are chosen uniformly at random among all items. In the experiments, we vary $|\mathcal{V}_k|$ from 50 up to 200. For the evaluation, we split the ratings into training and test sets, and we vary the ratio $|\mathcal{V}_k^t|/|\mathcal{V}_k|$ of training to total ratings from 30% up to 90%. Table 1 summarizes the parameters of our construction of group ratings.

The scores of the group ratings are assigned according to different strategies, resulting in 7 distinct datasets as detailed in the following.

AVERAGE. The rating a group g_k gives to an item v_j is equal to the average rating across the group members, i.e.,

$$r_{kj} = \frac{1}{|g_k|} \sum_{u_i \in g_k} r_{ij}.$$

This type captures the setting where all group members jointly and equally make a decision.

LEAST-MISERY. The rating of group g_k to item v_j is equal to the minimum rating among the group members, i.e.,

$$r_{kj} = \min_{u_i \in g_k} r_{ij}.$$

This models the case where the group behaves under a least-misery principle, so as not to displease any individual member.

DICTATOR. The rating of group g_k to item v_j is equal to the rating of one group member, chosen uniformly at random, i.e., $r_{kj} = r_{ij}$, where $i \sim unif[1, |g_k|]$. This type models the case where decisions in a group are governed by the desires of a single person, e.g., the boss of a company, the child in a family.

In the next four types, the rating of group g_k to item v_j is a weighted average rating across the group members. It is the definition of the weights that differs.

WEIGHTED-GLOBAL. Each user is assigned a uniformly random weight, and thus the group g_k rating to v_j is

$$r_{kj} = \frac{1}{\sum_i w_i} \sum_{u_i \in g_k} w_i r_{ij},$$

where $w_i \sim unif[0, 1]$. Note that the weight of a particular user *persists* across groups — hence the characterization global. This type assumes users have a consistent predefined behavior when in groups, e.g., always willing to compromise.

LEADER-GLOBAL. Each user u_i has a global weight either very small $w_i = 0.1$, or very large $w_i = 10$, where the latter is chosen with a probability $\frac{1}{|g_k|}$, so that in a group there is on average one

person with strong opinion. This captures the case where users are either leaders ($w_i = 10$) or followers ($w_i = 0.1$) when in groups.

WEIGHTED-LOCAL. Each group member is assigned a uniformly random weight, and thus the group rating is

$$r_{kj} = \frac{1}{\sum_i w_{ik}} \sum_{u_i \in g_k} w_{ik} r_{ij},$$

where $w_{ik} \sim unif[0, 1]$. The difference with WEIGHTED-GLOBAL is that users may have different weights across groups, capturing thus the case where users exhibit group-specific behavior.

LEADER-LOCAL. Each member u_i of a group g_k has a weight either very small $w_{ik} = 0.1$, or very large $w_{ik} = 10$, where the latter is chosen with a probability $\frac{1}{|g_k|}$. Here, a user may exhibit different bipolar behavior across groups, e.g., follower among co-workers, leader among friends.

4.1.3 Methods. We evaluate our proposed group rating prediction models, RESIDUAL and TRIAD, against variants of PROF-AGG and REC-AGG using average (avg), minimum (min), maximum (max), and product (prd) aggregation strategies, and the INF-MATCH method described in Section 2.3.

The PROF-AGG, REC-AGG, and RESIDUAL methods require a module to predict user-item ratings. In our implementation, we have used the matrix factorization model described in Section 2.2. The model parameters ($\mathbf{U}$ and $\mathbf{V}$) were learned using SGD, where the hyperparameters factor dimensionality ($f = 10$), regularization parameter ($\lambda = 0.005$), and learning rate of SGD ($\eta = 0.005$) were determined by cross validation. Moreover, the factorization of matrix Ξ in RESIDUAL was also determined by SGD ($f = 6$, $\lambda = 0.001$, $\eta = 0.025$). Similarly, parameters $\mathbf{U}$, $\mathbf{V}$, $\mathbf{b}$ of TRIAD were determined by SGD ($f = 10$, $\lambda'_{\mathbf{U}} = \lambda'_{\mathbf{V}} = \lambda'_{\mathbf{b}} = 0.001$, $\eta = 0.005$). Finally, the Poisson-mixture parameters of INF-MATCH were determined by the Expectation Maximization algorithm as discussed in [8].

4.1.4 Evaluation Metrics. Our proposed methods, similar to other matrix factorization techniques, are designed to minimize the prediction error of the group ratings. Hence the two main evaluation metrics we employ are mean square error variants. Nonetheless, to obtain a more general picture of performance, we also consider two ranking metrics. We note that the reported values of these metrics are the averages across at least 9 different runs, each with different train/test data splits and random seeds for the SGD.

RMSE. The Root Mean Square Error is computed as:

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{G}|} \sum_{g_k \in \mathcal{G}} \frac{1}{|\mathcal{V}_k^{ev}|} \sum_{v_j \in \mathcal{V}_k^{ev}} (r_{kj} - \hat{r}_{kj})^2},$$

where $\mathcal{V}_k^{ev}$ is the set of test (evaluation) items for group g_k. The metric captures the overall accuracy of the predicted group ratings; lower values are better.

M-RMSE. The maximum RMSE within a group is computed as:

$$\text{M-RMSE} = \max_{g_k \in \mathcal{G}} \sqrt{\frac{1}{|\mathcal{V}_k^{ev}|} \sum_{v_j \in \mathcal{V}_k^{ev}} (r_{kj} - \hat{r}_{kj})^2},$$

and indicates the worst-case accuracy across any group. Compared to RMSE, this metric better captures the robustness of the recommender system, as low values indicate that *all groups* will receive good recommendations.

The next two metrics measure the quality of the items' ranking induced by the predicted ratings. For group g_k, let $L_k = v_{j_{(1)}}, v_{j_{(2)}}, \ldots$ denote the list of test items in $\mathcal{V}_k^{ev}$ ranked decreasingly be their predicted group rating $\hat{r}_{kj_{(i)}}$.

NDCG@N. The Discounted Cumulative Gain (DCG) at rank N for group g_k is:

$$\text{DCG}_k@N = r_{kj_{(1)}} + \sum_{i=2}^{N} \frac{r_{kj_{(i)}}}{\log(i+1)}.$$

The Ideal Discounted Cumulative Gain (IDCG) is defined as the maximum possible DCG, achieved when the items are ranked decreasingly by ther actual group rating. The Normalized Discounted Cumulative Gain at rank N is then computed as the average ratio of DCG over IDCG across all groups:

$$\text{NDCG@}N = \frac{1}{|\mathcal{G}|} \sum_{g_k \in \mathcal{G}} \frac{\text{DCG}_k@N}{\text{IDCG}_k@N}.$$

NDCG takes values in the range $[0, 1]$, where higher values are better.

The last metric measures the quality of the ranking with respect to their relevance. For this reason, we must introduce a relevance criterion. Specifically, we treat a test set item as *relevant* when its actual group rating is greater than a threshold ρ. For group g_k, let $L_k^{rel} = v_{j_{[1]}}, v_{j_{[2]}}, \ldots$ denote the list of *relevant* test items in $\mathcal{V}_k^{ev}$ ranked decreasingly be their *actual* group rating $r_{kj_{[i]}}$. Note the distinction between the i-th ranked relevant item $v_{j_{[i]}}$ according to its actual rating and the i-th ranked item $v_{j_{(i)}}$ according to its predicted rating.

MAP. The Mean Average Precision is

$$\text{MAP} = \frac{1}{|\mathcal{G}|} \sum_{g_k \in \mathcal{G}} \frac{1}{|L_k^{rel}|} \sum_{i=1}^{|L_k^{rel}|} \frac{i}{rank_k(v_{j_{[i]}})},$$

where $rank_k(v_{j_{[i]}})$ is the rank of item $v_{j_{[i]}}$ in the list L_k (which is sorted according to predicted ratings). MAP takes values in the range $[0, 1]$, where higher values are better.

4.2 Results

4.2.1 Real Dataset. Table 2 presents the metric values for all methods. In the first two columns where prediction error is measured, lower values are better. In the last three columns showing ranking performance, higher values are better. In each column we mark the best obtained value in bold. Note that INF-MATCH produces only a ranked list of items and cannot predict ratings; thus we cannot compute its prediction error.

We should note that predicting the behavior of groups in REAL is a difficult task, as also observed in [5]. As a general conclusion, all metric values for all methods are considerably worse than their counterparts in the synthetically generated datasets. To some extent, this can be attributed to the fact that this is a small dataset involving few items that more or less are all equally preferable. Our methods in particular are hindered by two additional facts, that there are only few group ratings to learn from, and that users are not shared among groups.

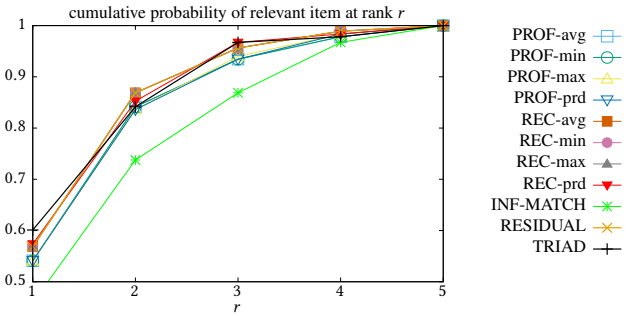


Figure 1: Chance of relevant item among first r results in REAL

Table 2: Metrics for REAL

	RMSE	M-RMSE	NDCG@3	NDCG@5	MAP
PROF-avg	1.041	**1.814**	0.785	0.676	0.218
PROF-min	0.919	2.170	0.772	0.682	0.209
PROF-max	1.766	2.518	0.788	0.682	0.218
PROF-prd	1.311	2.415	0.779	0.682	0.218
REC-avg	1.007	1.955	0.829	**0.791**	0.227
REC-min	0.974	1.894	0.829	**0.791**	0.227
REC-max	1.109	2.609	0.829	**0.791**	0.227
REC-prd	1.947	3.698	**0.844**	0.640	0.232
INF-MATCH	—	—	0.635	0.498	0.178
RESIDUAL	**0.822**	2.172	0.829	**0.791**	0.227
TRIAD	0.837	2.137	0.828	0.676	**0.241**

Nonetheless, it is important to notice that our proposed methods achieve their goal of minimizing the prediction error of the group ratings, as they have by far the lowest RMSE values. Looking at the M-RMSE column, we note however that there is some variance in the prediction error across groups. In particular, there exist a few groups for which our method did not have the lowest prediction error. For these groups, it turns out that averaging their profiles (PROF-avg) was a better approach.

To assess ranking quality, we set the relevance threshold to 3, meaning that only the top-2 destinations chosen by the groups are considered relevant. We observe that our methods have a good performance but not always the best. TRIAD achieves the best MAP, while RESIDUAL has the second best NDCG at rank 3 and the best at rank 5. We also note that among the existing methods, aggregating the predictions (REC-AGG) was a better approach than aggregating profiles (PROF-AGG) with respect to ranking evaluation metrics.

As there is at most two relevant items in (the test subset of) REAL, we also investigated how far in the ranked list compiled by each recommender would we have to go in order to see the first relevant item. Figure 1 plots the cumulative probability of seeing the first relevant item at each rank. All methods returned a relevant item in their first 5 positions, and thus their cumulative probability at rank 5 is 1. With the exception of INF-MATCH, all methods exhibit similar performance. Note that TRIAD has the highest chance of returning a relevant item as the first result (60.1%). TRIAD has also the highest chance (tied with REC-prd) of returning a relevant item among the top-3 (96.7%), while RESIDUAL has the highest chance (tied with REC-{avg, min, max}) for returning a relevant item among the top-2 (86.9%) and among the top-4 (98.9%).

4.2.2 Synthetic Datasets. In the first round of experiments, we investigate the performance of all methods in the standard scenario, i.e., when all synthetic group ratings parameters are set to their default values (see Table 1). Tables 3 through 8 summarize the quality of the group recommendation for six of the synthetic datasets; AVERAGE is omitted due to lack of space.

Table 3: Metrics for LEAST-MISERY

	RMSE	M-RMSE	NDCG@5	NDCG@10	MAP
PROF-avg	1.762	2.803	0.974	0.974	**0.889**
PROF-min	1.763	2.792	0.973	0.973	0.887
REC-avg	0.469	1.053	0.968	0.970	0.871
REC-min	0.451	1.070	0.970	0.971	0.866
RESIDUAL	0.443	1.081	**0.975**	**0.974**	0.888
TRIAD	**0.389**	**1.016**	0.972	0.972	0.884

We note that we have excluded the prd and max variants of the PROF-AGG and REC-AGG methods due to their poor performance (see Table 2 for REAL), especially in the prediction error metrics (RMSE and M-RMSE). Although their ranking quality was good, it was never better that the avg and min variants. For similar reasons, we have also excluded INF-MATCH; e.g., in one setting its MAP was at about 0.4, while all others were above 0.9.

Overall, we make the following important observations. First, in all strategies and under all metrics (except for two cases under MAP), our models are the best methods, often by far.

Second, regarding prediction error, RESIDUAL and TRIAD are much more accurate than existing methods, particularly so in the four weighted average datasets (WEIGHTED and LEADER variants). This is to be expected, since our methods are explicitly designed to learn the best way to linearly combine individual ratings. In almost all other settings, they are the two best methods. Even in their least favorable datasets (LEAST-MISERY and DICTATOR), where group ratings are not linear combinations of user ratings, they have a clear benefit over the second best. Note that TRIAD is always the best method under the RMSE metrics, and is thus the recommended approach when prediction error matters.

Third, with respect to the ranking metrics, RESIDUAL and TRIAD are still the best methods (except these two MAP cases) but by a smaller margin. This is to be expected, as they are explicitly designed to optimize for prediction error instead. Note that RESIDUAL in LEAST-MISERY is the best method under NDCG, and second best under MAP. Despite the solid performance of our methods, we see an opportunity in designing group recommenders explicitly targeting ranking quality.

Fourth, existing approaches cannot take advantage of the group rating history and have thus relatively poor performance, except in extreme cases that are tailor-made for them, namely AVERAGE for REC-avg, and LEAST-MISERY for REC-min. Overall the REC-AGG variants have significantly lower prediction error in all strategies considered, but among them REC-avg is the winner. This is a non-surprising observation that corroborates the fact that averaging works well in most cases [14]. On the other hand, with respect to ranking quality, the PROF-AGG variants perform marginally better in some strategies, and in two cases are even the best methods.

In the second round of experiments, we study the sensitivity of all methods as we vary the synthetic group ratings parameters. Figures 2 through 3 present the results of our study. It is clear that our models are robust and remain the best under all examined settings. On the other hand, the performance of existing approaches, and particularly of the PROF-AGG variants, varies significantly.

Table 4: Metrics for DICTATOR

	RMSE	M-RMSE	NDCG@5	NDCG@10	MAP
PROF-avg	1.103	1.964	0.782	0.532	0.941
PROF-min	1.087	2.009	0.782	0.514	0.936
REC-avg	0.499	1.122	0.780	0.531	0.934
REC-min	0.705	1.723	0.779	0.526	0.934
RESIDUAL	0.406	0.739	0.779	0.503	0.932
TRIAD	**0.347**	**0.726**	**0.789**	**0.547**	**0.941**

Table 5: Metrics for WEIGHTED-GLOBAL

	RMSE	M-RMSE	NDCG@5	NDCG@10	MAP
PROF-avg	1.157	2.167	0.976	0.98	0.913
PROF-min	1.152	2.205	0.975	0.979	0.911
REC-avg	0.378	0.6	0.975	0.979	0.914
REC-min	0.593	1.169	0.976	0.979	0.909
RESIDUAL	0.362	0.564	0.977	0.979	0.914
TRIAD	**0.319**	**0.478**	**0.979**	**0.982**	**0.923**

Table 6: Metrics for LEADER-GLOBAL

	RMSE	M-RMSE	NDCG@5	NDCG@10	MAP
PROF-avg	1.249	2.155	0.977	0.984	0.935
PROF-min	1.244	2.167	0.977	0.984	**0.942**
REC-avg	0.425	0.872	0.975	0.981	0.929
REC-min	0.636	1.373	0.975	0.982	0.931
RESIDUAL	0.381	0.674	0.977	0.984	0.939
TRIAD	**0.344**	**0.633**	**0.981**	**0.987**	0.94

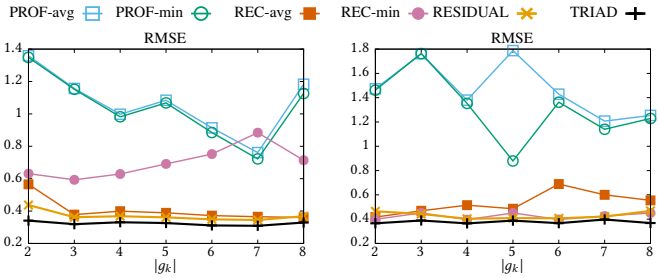
Table 7: Metrics for WEIGHTED-LOCAL

	RMSE	M-RMSE	NDCG@5	NDCG@10	MAP
PROF-avg	1.129	2.119	0.974	0.977	0.9
PROF-min	1.122	2.147	0.974	0.977	0.896
REC-avg	0.404	0.714	0.97	0.974	0.897
REC-min	0.591	1.204	0.972	0.976	0.896
RESIDUAL	0.537	1.426	0.973	0.977	0.898
TRIAD	**0.341**	**0.614**	**0.975**	**0.978**	**0.904**

Table 8: Metrics for LEADER-LOCAL

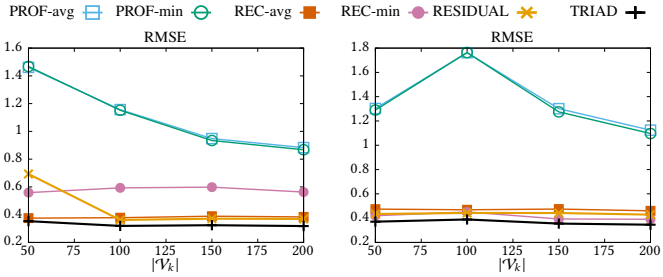
	RMSE	M-RMSE	NDCG@5	NDCG@10	MAP
PROF-avg	0.946	1.908	0.973	0.977	0.877
PROF-min	0.941	1.91	0.972	0.976	0.878
REC-avg	0.416	0.902	0.974	0.978	0.892
REC-min	0.64	1.383	0.973	0.976	0.885
RESIDUAL	0.374	0.549	0.973	0.977	0.888
TRIAD	**0.331**	**0.480**	**0.977**	**0.980**	**0.897**

In the last round of experiments, we investigate the performance under a cold-start scenario. As before, we consider 50 groups populated with 100 distinct users. For 10 of these groups, we assign profiles under the WEIGHTED-GLOBAL scheme that have zero group ratings (extreme cold) up to 30 ratings (warm). We then ask the recommender to provide predictions for 100 items for these 10 groups and measure the prediction error; results are shown in Figure 4. Note that the behavior of REC-avg is the same regardless of the size of the profiles; small variations are due to randomness. In the extreme case of empty group profiles, our methods exhibit significant prediction error. Clearly, REC-avg should be the method of choice for such situations. However, the important thing to notice is that as the group profiles increase in size, our methods, TRIAD particularly, are able to quickly reduce the prediction error. When only 5 group ratings are available, TRIAD achieves comparable RMSE to REC-avg, while it reduces the maximum RMSE across



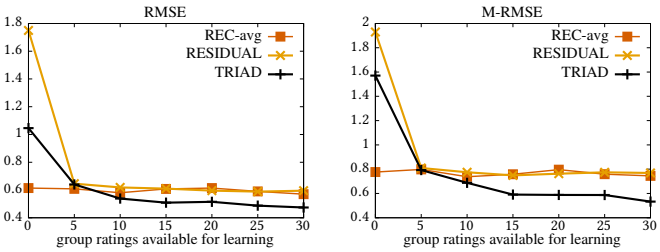
(a) WEIGHTED-GLOBAL (b) LEAST-MISERY

Figure 2: RMSE vs number of users per group



(a) WEIGHTED-GLOBAL (b) LEAST-MISERY

Figure 3: RMSE vs number of ratings per group



(a) RMSE (b) M-RMSE

Figure 4: Prediction error for cold-start groups

groups (M-RMSE). As the profile size increases, TRIAD further reduces the variance of RMSE among groups.

5 CONCLUSIONS

This work proposes two group recommenders that explicitly model the difference in the behavior of users when they are members of a group and individually. An experimental study with real and synthetic group ratings demonstrates the superiority of our proposed methodology according to all evaluation metrics studied. In particular, the TRIAD model, which explicitly learns the behavior of users in groups, is the best method in the large majority of the experiments. The RESIDUAL model is often the second best method, and in some settings under ranking evaluation metrics, is even the best. We also find that aggregating individual recommendations generally provides better recommendations for the group compared to when constructing an aggregate group profile.

ACKNOWLEDGMENTS

The author would like to thank Amra Delic for compiling the REAL dataset.

REFERENCES

[1] Liliana Ardissono, Anna Goy, Giovanna Petrone, Marino Segnan, and Pietro Torasso. 2003. Intrigue: Personalized Recommendation of Tourist Attractions for Desktop and Hand Held Devices. *Applied Artificial Intelligence* 17, 8-9 (2003), 687–714. DOI: http://dx.doi.org/10.1080/713827254

[2] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group recommendations with rank aggregation and collaborative filtering. In *RecSys*. 119–126. DOI: http://dx.doi.org/10.1145/1864708.1864733

[3] Shlomo Berkovsky and Jill Freyne. 2010. Group-based recipe recommendations: analysis of data aggregation strategies. In *RecSys*. 111–118. DOI: http://dx.doi.org/10.1145/1864708.1864732

[4] Andrew Crossen, Jay Budzik, and Kristian J. Hammond. 2002. Flytrap: intelligent group music recommendation. In *IUI*. 184–185. DOI: http://dx.doi.org/10.1145/502716.502748

[5] Amra Delic, Julia Neidhardt, Thuy Ngoc Nguyen, Francesco Ricci, Laurens Rook, Hannes Werthner, and Markus Zanker. 2016. Observing Group Decision Making Processes. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*. 147–150. DOI: http://dx.doi.org/10.1145/2959100.2959168

[6] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. 2001. Rank aggregation methods for the Web. In *WWW*. 613–622. DOI: http://dx.doi.org/10.1145/371920.372165

[7] 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 2010 International ACM SIGGROUP Conference on Supporting Group Work, GROUP 2010, Sanibel Island, Florida, USA, November 6-10, 2010*, Wayne G. Lutters, Diane H. Sonnenwald, Tom Gross, and Madhu Reddy (Eds.). ACM, 97–106. DOI: http://dx.doi.org/10.1145/1880071.1880087

[8] Jagadeesh Gorla, Neal Lathia, Stephen Robertson, and Jun Wang. 2013. Probabilistic group recommendation via information matching. In *WWW*. 495–504. http://dl.acm.org/citation.cfm?id=2488432

[9] Anthony Jameson and Barry Smyth. 2007. Recommendation to Groups. In *The Adaptive Web, Methods and Strategies of Web Personalization*, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Springer, 596–627.

[10] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*. 426–434. DOI: http://dx.doi.org/10.1145/1401890.1401944

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (Aug. 2009), 30–37. DOI: http://dx.doi.org/10.1109/MC.2009.263

[12] Qiuxia Lu, Diyi Yang, Tianqi Chen, Weinan Zhang, and Yong Yu. 2011. Informative household recommendation with feature-based matrix factorization. In *CAMRa*. ACM, 15–22.

[13] Judith Masthoff. 2004. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Model. User-Adapt. Interact.* 14, 1 (2004), 37–85. DOI: http://dx.doi.org/10.1023/B:USER.0000010138.79319.fd

[14] Judith Masthoff. 2015. Group Recommender Systems: Aggregation, Satisfaction and Group Attributes. In *Recommender Systems Handbook*. 743–776. DOI: http://dx.doi.org/10.1007/978-1-4899-7637-6_22

[15] Judith Masthoff and Albert Gatt. 2006. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Model. User-Adapt. Interact.* 16, 3-4 (2006), 281–319. DOI: http://dx.doi.org/10.1007/s11257-006-9008-3

[16] Joseph F. McCarthy and Theodore D. Anagnost. 1998. MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW '98)*. ACM, New York, NY, USA, 363–372. DOI: http://dx.doi.org/10.1145/289444.289511

[17] Kevin McCarthy, Maria Salamó, Lorcan Coyle, Lorraine McGinty, Barry Smyth, and Paddy Nixon. 2006. CATS: A Synchronous Approach to Collaborative Group Recommendation. In *FLAIRS*. 86–91. http://www.aaai.org/Library/FLAIRS/2006/flairs06-015.php

[18] Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. 2001. PolyLens: A recommender system for groups of user. In *ECSCW*. 199–218.

[19] Alan Said, Shlomo Berkovsky, Ernesto William De Luca, and Jannis Hermanns. 2011. Challenge on context-aware movie recommendation: CAMRa2011. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius (Eds.). ACM, 385–386. DOI: http://dx.doi.org/10.1145/2043932.2044015

[20] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*. 1257–1264. http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization

[21] Lara Quijano Sánchez, Juan A. Recio-García, Belén Díaz-Agudo, and Guillermo Jiménez-Díaz. 2013. Social factors in group recommender systems. *ACM TIST* 4, 1 (2013), 8:1–8:30. DOI: http://dx.doi.org/10.1145/2414425.2414433

[22] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295. DOI: http://dx.doi.org/10.1145/371920.372071

[23] Christophe Senot, Dimitre Kostadinov, Makram Bouzid, Jérôme Picault, and Armen Aghasaryan. 2011. Evaluation of Group Profiling Strategies. In *IJCAI*. 2728–2733. http://ijcai.org/papers11/Papers/IJCAI11-454.pdf

[24] Christophe Senot, Dimitre Kostadinov, Makram Bouzid, Jérôme Picault, Armen Aghasaryan, and Cédric Bernier. 2010. Analysis of Strategies for Building Group Profiles. In *User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings (Lecture Notes in Computer Science)*, Paul De Bra, Alfred Kobsa, and David N. Chin (Eds.), Vol. 6075. Springer, 40–51. DOI: http://dx.doi.org/10.1007/978-3-642-13470-8_6

[25] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. 2006. TV Program Recommendation for Multiple Viewers Based on user Profile Merging. *User Model. User-Adapt. Interact.* 16, 1 (2006), 63–82. DOI: http://dx.doi.org/10.1007/s11257-006-9005-6