# SQUIRREL 2.0: Fairness & Explanations for Sequential Group Recommendations

Md. Mahade Hasan[1], Soha Pervez[1], Maria Stratigi[1] and Kostas Stefanidis[1]

[1]Tampere University, Tampere, Finland

**Abstract**

A growing number of applications enable users to form groups for activities, like visiting a restaurant or watching a movie, making group recommenders more prevalent than ever. SQUIRREL is a framework for sequential group recommendations, providing a different recommendation in each round. It relies on Reinforcement Learning to select appropriate group recommendation algorithms based on the current state of the group. At each round of recommendations, it calculates the satisfaction of each group member and selects a recommendation method that will produce the maximum reward. In this paper, we incorporate two new reward functions, utilizing the *m-proportionality* measure to produce recommendations that are fairer to the group by promoting at least *m* items in the group recommendation list that each member prefers. Moreover, we study a user case explaining the SQUIRREL recommendations.

**Keywords**

Group recommendations, Sequential recommendations, Fairness, Explanations

## 1. Introduction

The vast expansion of social media and the ease of communication have encouraged more people to socialize and engage in group activities. This has led to the group recommendation research area becoming one of the more popular ones for recommender systems. Group recommendations should be able to recommend items relevant to the group members without any bias, meaning that a user should have at least one item in the group recommendation list that is relevant to them. In addition, the system needs to take into account previous interactions with the group. This type of system is known as a sequential group recommender, which ensures that, after a sequence of recommendations, all members are satisfied with the suggestions and no one is discriminated against.

Group recommendations can be approached in many ways, but it is not always clear which approach is best for each situation. Due to the wide range of group recommendation methods and their widely differing approaches, evaluating all of them and deciding which one works best for each test scenario is difficult and time-consuming. Not all group recommendation methods have the same performance when utilized in different domains. For example, a group recommender system build for movie recommendations has a vastly different performance when it is used in different domains [1]. Because of this, transferring a recommender system to another domain is not an easy task.

The SQUIRREL framework [2] provides a solution to this problem based on reinforcement learning techniques, which is an ideal model since it directly mirrors the sequential nature of recommendations. The model is composed of three main elements: state, actions, and reward. The state of the model is how satisfied the group members are with all the previous recommendation rounds. Actions are the different group recommendation methods that the model has available. Rewards represent how successful the action chosen was. At each recommendation round, SQUIRREL automatically selects the best group recommendation method to apply (action) based on how satisfied the group members are (state).

In this work, we mainly focus on achieving fair sequential group recommendations. We facilitate this by utilizing reward functions that promote fairness among the group. We aim to maximize the satisfaction of the group members, while trying to minimize their disagreement. We define a user's satisfaction by how relevant the recommended items are to each user and their disagreement as the difference between their satisfaction and the highest satisfied member. In addition, we examine the performance of the model when utilizing the *m-proportionality* measure [3] that considers the recommended list fair when users like at least *m* items on it.

Finally, we not only want our model to produce fair recommendations but also to be able to explain why these suggestions were made. Since the group recommendation process is primarily treated as a black box, explanations can increase users' trust in the system. In this work, we demonstrate how to explain the SQUIRREL recommendations throughout multiple recommendation rounds, using the historical data of the group.

*DOLAP 2024: 26 International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data, co-located with EDBT/ICDT 2024, March 25, 2024, Paestum, Italy*

mdmahade.hasan@tuni.fi (Md. M. Hasan); soha.pervez@tuni.fi (S. Pervez); maria.stratigi@tuni.fi (M. Stratigi); konstantinos.stefanidis@tuni.fi (K. Stefanidis)

## 2. SQUIRREL for Group Recommendations

Let $I$ be a set of data items and $U$ a set of users. $G$ denotes a group of users where $G \subseteq U$. For each user $u$ in $G$, $B_u^j$ is the list of recommended items, as a single recommender system has generated them at a recommendation round $j$ for $u$. At round $j$, SQUIRREL chooses an appropriate aggregation function based on the current state of the group, to combine the individual members' recommendation lists $B_u^j$ into one group recommendation list $GL_G^j$.

SQUIRREL can be described as a Markov decision process, where an agent interacts with an environment $E$ to maximize the accumulative reward after each recommendation round. The Markov decision process can be described by a tuple of $(S, \mathrm{A}, P, R)$. The goal of the model is to find a policy $\pi(a|s)$ that takes action $a \in A$ during state $s \in S$ to maximize the expected discounted cumulative reward after $\mu$ recommendation rounds: $max\,\mathbb{E}[R(\mu)]$, where $R(\mu) = \sum_{t=0}^{\mu} \gamma R_\alpha(s, s')$, with $0 \leq \gamma \leq 1$.

***S*** is a continuous space that describes the environmental state, i.e., the group state, that is, how satisfying the recommended items were for each group member. A user $u$'s satisfaction with the group recommendation list $sat(u, GL_G^j)$ is calculated by comparing how relevant the items recommended to the group were with the most relevant items for that user. That is: $sat(u, GL_G^j) = \frac{\sum_{d \in GL_G^j} p_j(u,d)}{\sum_{d \in B_{u,k}^j} p_j(u,d)}$, where $p_j(u, d)$ returns the predicted score of item $d$ for user $u$ at recommendation round $j$, as it was produced by a single recommender.

The group state is calculated based on how satisfied each of its members is in overall, that is, their average satisfaction up to the current recommendation round. Formally: $satO(u, \mathbb{GR}) = \frac{\sum_{j=1}^{\mu} sat(u, GL_G^j)}{\mu}$.

In turn, ***A*** is a set of distinct actions consisting of SQUIRREL aggregation functions. We employ 6 different methods as presented in [2], namely: *Average*, *RP80 [4]*, *Par [5]*, *SDAA [6]*, *SIAA [6]* and *Avg+ [6]*. $P_a(s, s')$ defines the probability to transition from state $s$ to state $s'$ during round $j$ under the action $a$. Formally, $P_a(s, s') = Pr(s_{j+1} = s'|s_j = s, a_j = a)$. Finally, $R_a(s, s')$ is the reward gained from transitioning from state $s$ to state $s'$. The reward describes the quality of recommendations given by the model. The model determines if an action is appropriate based on the reward that it receives by taking the action.

### 2.1. Satisfaction & Disagreement Rewards

One option to define the reward is via the group satisfaction score. This score will indicate how well the system is able to balance the individual needs of the group members. Specifically, we define the group $G$'s satisfaction concerning a group recommendation list $GL_G^j$ to be the average of the satisfaction scores in the group: $groupSat(GL_G^j) = \frac{\sum_{u \in G} sat(u, GL_G^j)}{|G|}$. Subsequently, we define the overall group satisfaction of a group $G$ for a recommendation sequence $\mathbb{GR}$ of $\mu$ group recommendations as: $groupSatO(\mathbb{GR}) = \frac{\sum_{u \in G} satO(u, \mathbb{GR})}{|G|}$.

The overall group satisfaction can be used as an expression of the reward achieved by action $a$ in recommendation round $j$, that is: $R_s(\mathbb{GR}^j) = groupSatO(\mathbb{GR}^j)$, where $\mathbb{GR}^j$ refers to all the rounds up to the j$^{th}$ one.

However, this reward may lead us to somehow ignore the dissatisfaction of a user, since it considers only the average of the group members' individual satisfaction scores. This observation leads us to a new utility score that considers the user disagreement. We utilize the harmonic mean of overall group satisfaction $groupSatO$ and overall group disagreement $groupDisO$, which is the difference between the maximum and minimum overall satisfaction among the group members: $groupDisO(\mathbb{GR}) = max_{u \in G} satO(u, \mathbb{GR}) - min_{u \in G} satO(u, \mathbb{GR})$. This measure is often referred to as the F-score. We simulate the group agreement using $1 - groupDisO$, considering the input functions needed by the F-score.

$$R_{sd}(\mathbb{GR}^j) = 2\frac{groupSatO(\mathbb{GR}^j) * (1 - groupDisO(\mathbb{GR}^j))}{groupSatO(\mathbb{GR}^j) + (1 - groupDisO(\mathbb{GR}^j))} \quad (1)$$

This reward function reflects the degree of preference for an item among the group members as well as the level of disagreement or agreement between members.

### 2.2. Fairness Reward

To form a group recommendation list, $GL_G^j$, that is fair to every member of the group, we consider one fairness criterion called proportionality [3]. The fairness concept of fair division of resources inspires this function. The main idea behind proportionality is to count for each group member, the items that are present in the group recommendation list that they prefer. A user likes an item, if it is in the top $\Delta$% of the user's individual recommendation list, $B_u^j$, as it was produced by a single recommender system. When a recommendation list contains at least m ($m \geq 1$) items that a user likes, then we call it m-proportional. If $m = 1$, we call this list single-proportional, otherwise referred to as multi-proportional.

The m-proportional approach creates mutual acceptance of the recommended items among the group members. When a list contains at least m items that a user strongly prefers, that user is likely to be more accepting of other items in the list that they may not like. This tolerance is based on the understanding that other group members may prefer those items. Then, m-proportionality can be defined as follows.

**[m-PROPORTIONALITY]** Given a group G, and group recommendation list $GL_G^j$, We express the m-proportionality as: $R_{mprop}(G, GL_G^j) = \frac{|G_p|}{|G|}$, where $G_p \subseteq G$ represents the set of users within the group for which $GL_G^j$ is m-proportional.

We utilize the m-proportionality fairness measure, to introduce another reward function, which is the harmonic mean of the overall group satisfaction and m-proportionality. Formally:

$$R_{smprop}(G, GL_G^j) = 2\frac{groupSatO(G, GL_G^j) * R_{mprop}(G, GL_G^j)}{groupSatO(G, GL_G^j) + R_{mprop}(G, GL_G^j)} \quad (2)$$

This measure combines the overall satisfaction of a group along with m-proportionality fairness. In this reward function, the goal is to ensure all members are recommended some of their preferred items while maintaining the group's overall satisfaction.

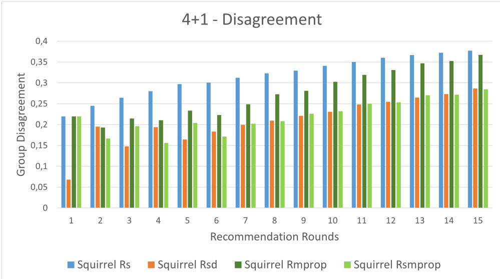
## 3. Evaluation

To evaluate our work, we use the 20M MovieLens dataset and divide it into two parts [2]. The system initiates with the first part which contains 60% of the movies to avoid the cold start problem. The latter portion is divided into 14 chunks and used for the experiments. We use (4+1) groups that contain 4 similar and 1 dissimilar user, counting similarity using the Pearson Correlation function. At each round, we suggest a set of 10 items to the group, without suggesting those items again. We generated 100 different groups: 80 groups were used for training and 20 for testing. For $R_{mprop}$ and $R_{smprop}$ we chose $m = 2$ and $\Delta = 20\%$.

### 3.1. Experimental Results

When evaluating the outcomes for the 4+1 training set across the four reward functions, we observe in Figure 1 that the $R_s$ reward function yields the best group satisfaction scores (Figure 1(a)). On the other hand, $R_{mprop}$ comes up with the lowest value. This is because the $R_s$ reward function aims to maximize overall group satisfaction and $R_{mprop}$ tries to recommend at least a fixed number of preferred items per user. In our current settings, it is hard for a group of five people to retain two items from a list of ten items because there is at least one member who stands out from the rest. That means it is difficult to satisfy every group member. When satisfaction is included with m-proportionality in $R_{smprop}$ it shows better satisfaction. However, note that $R_s$ and $R_{mprop}$ result in higher group disagreement compared to the $R_{sd}$ and $R_{smprop}$ reward functions (Figure 1(b)). Also, $R_{smprop}$ produces slightly lower disagreement scores compared to $R_{sd}$. This outcome matches our expectations since $R_{smprop}$ takes into account both group satisfaction and m-proportionality in its calculation which results in a balanced set of recommended items.


(a)



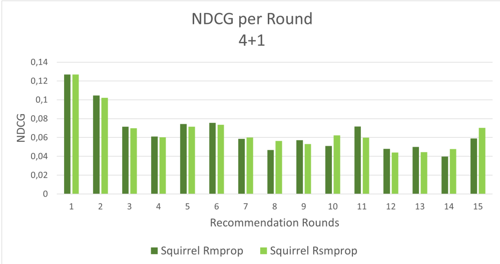

(b)

**Figure 1:** Group Satisfaction and Disagreement for $SQUIRREL - R_s$, $R_{sd}$, $R_{mprop}$, $R_{smprop}$ in 4+1 groups.

**Table 1**
NDCG values for the SQUIRREL models in all test scenarios.

| | $R_s$ | $R_{sd}$ | $R_{mprop}$ | $R_{smprop}$ |
|---|---|---|---|---|
| 4+1 | 0.052 | 0.054 | 0.066 | 0.067 |

We also calculate the Normalized Discounted Cumulative Gain (NDCG) value to analyze the quality of the methods. NDCG shows how many items in the group recommended list are also present in the user preferences list (Table 1). The table showcases the average scores calculated after all 15 rounds of recommendations. From the table, we can see that the new fairness-based SQUIRREL models $R_{mprop}$ and $R_{smprop}$ have higher values compared to others. This means it can identify more relevant items for the groups. In Figure 2, we plotted the NDCG scores for the two new models. There, it becomes evident that in the initial round of recommendations, these methods show a higher performance, which varies in the subsequent rounds. The reason for this is two-fold. First, due to the varied sparsity of the dataset as each chunk is added to the system. Second, after each recommendation round, the top 10 items for each group are excluded from




(a)

**Figure 2:** NDCG Values per recommendation round for $SQUIRREL - R_{mprop}$, $R_{smprop}$ in the 4+1 test scenario.

consideration, contributing to the performance decline.

## 4. Explanations for SQUIRREL

Typically, explanations results in the users becoming more trusting in the system, which makes the system more persuasive and effective, resulting in a higher level of satisfaction for the users. For our work, we have developed why questions and their respective explanations. We explore genres of movies and how many times they have been recommended to the group as a whole as well as to individual users of the group. We format our question as follows: *Why 'selected genre' occur 'selected frequency'?*

There are 19 genres and 3 frequencies to choose from: *selected genre = [ 'Action', 'Adventure', 'Animation', 'Children', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War', 'Western', 'IMAX' ],*
*selected frequency = [ 'a few times', 'many times', 'not at all' ].*

Each group will have the freedom of choice about which genre to inquire about. For example, if a group wants to ask a question related to the 'Musical' genre and why it appears 'a few times', the question becomes: *Why 'Musical' occurs 'a few times'?.*

We explain the 'Why' question in terms of a general explanation that is based on single-user recommendation lists for the users of the group, along with a model-based explanation that incorporates the summarized version of different aggregation methods. An example of such an explanation is the following: *The genre Musical is less likely to be enjoyed by 4 members of this group, therefore it occurs less frequently.*
*This explanation uses the SDAA action that balances the average predicted score of an item for the group with the predicted score of the least satisfied member.*

Finally, to facilitate easy comprehension of recommendations by the users of the group, we have produced visualizations showing:
*1. Group recommendations with a satisfaction score.*
*2. Group recommendations with disagreement score.*
*3. Single-user recommendations for all the users of a group.*

## 5. Related Work

For producing group recommendations (e.g., [4, 7, 8]), we employ a standard single-user recommender, apply it to each individual group member, and aggregate the group members lists into one single group recommendation list. In [9], fairness is presented as a constrained optimization problem. For a given set of rankings, the method provides the most similar ranking to the provided sets that satisfy a specific fairness requirement. For assessing fairness, [5] measures the degree of satisfaction for each group member with the group recommendation list, based on the relevance of the recommended items for each member. Differently, [10] uses the position of the items in the group recommendation list, exploiting the concept of Pareto optimality. More recently, [3] counts fairness using *proportionality*: when the user $u$ likes at least $m$ items in the recommended list, the user considers the list fair for them. In our work, we take inspiration from various fairness definitions and apply a comparable one using the users' satisfaction and disagreement scores. Additionally, we use directly the proportionality fairness measures as a reward function for our model.

In recent years, Reinforcement Learning (RL) has become increasingly popular in recommendation systems [11]. For example, [12] proposes an online personalized news recommendation framework based on DQNs, while [13] optimizes recommendation models for long-term accuracy using RL techniques. [14] incorporates randomness for fairness throughout, using Variational Autoencoders (VAEs), and penalizes items based on their historical popularity to promote diversity and minimize bias [15, 16, 17]. In our work, we aim to be more versatile regarding the domains in which our solution can be utilized and can incorporate a variety of strategies to compensate for the limitations inherent to every recommendation method.

## 6. Summary

In this work, we present a new reward function for the SQUIRREL model, aiming to increase the fairness of the sequential group recommendations. Using $m-proportionality$ produces better results in terms of group members' satisfaction and disagreement. In addition, we showcase how to provide explanations for recommendations produced by SQUIRREL.

## References


[1] R. D. Burke, M. Ramezani, Matching recommendation technologies and domains, in: F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 367–386.

[2] M. Stratigi, E. Pitoura, K. Stefanidis, Squirrel: A framework for sequential group recommendations through reinforcement learning, Inf. Syst. 112 (2023). URL: https://doi.org/10.1016/j.is.2022.102128. doi:10.1016/j.is.2022.102128.

[3] D. Serbos, S. Qi, N. Mamoulis, E. Pitoura, P. Tsaparas, Fairness in package-to-group recommendations, in: WWW, 2017.

[4] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, C. Yu, Group recommendation: Semantics and efficiency, PVLDB 2 (2009) 754–765.

[5] L. Xiao, Z. Min, Z. Yongfeng, G. Zhaoquan, L. Yiqun, M. Shaoping, Fairness-aware group recommendation with pareto-efficiency, in: RecSys, 2017.

[6] M. Stratigi, E. Pitoura, J. Nummenmaa, K. Stefanidis, Sequential group recommendations based on satisfaction and disagreement scores, Journal of Intelligent Information Systems (2021).

[7] L. Baltrunas, T. Makcinskas, F. Ricci, Group recommendations with rank aggregation and collaborative filtering, in: RecSys, 2010.

[8] E. Ntoutsi, K. Stefanidis, K. Nørvåg, H.-P. Kriegel, Fast group recommendations by applying user clustering, in: International conference on conceptual modeling, Springer, 2012, pp. 126–140.

[9] C. Kuhlman, E. Rundensteiner, Rank aggregation algorithms for fair consensus, Proceedings of the VLDB Endowment 13 (2020).

[10] D. Sacharidis, Top-n group recommendations with fairness, in: SAC, 2019.

[11] M. M. Afsar, T. Crump, B. Far, Reinforcement learning based recommender systems: A survey, 2021. arXiv:2101.06286.

[12] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, Z. Li, Drn: A deep reinforcement learning framework for news recommendation, in: WWW, 2018.

[13] L. Huang, M. Fu, F. Li, H. Qu, Y. Liu, W. Chen, A deep reinforcement learning based long-term recommender system, Knowledge-Based Systems (2021).

[14] R. Borges, K. Stefanidis, Enhancing long term fairness in recommendations with variational autoencoders, in: MEDES, 2019.

[15] R. Borges, K. Stefanidis, On mitigating popularity bias in recommendations via variational autoencoders, in: SAC, 2021.

[16] R. Borges, K. Stefanidis, F2VAE: a framework for mitigating user unfairness in recommendation systems, in: SAC, 2022.

[17] R. Borges, K. Stefanidis, On measuring popularity bias in collaborative filtering data, in: EDBT/ICDT Workshops, 2020.