Md. Mahade Hasan

FAIRNESS IN SEQUENTIAL GROUP RECOMMENDATIONS

Faculty of Information Technology and Communication Sciences (ITC)
Master's thesis
April 2024

Abstract

Md. Mahade Hasan: FAIRNESS IN SEQUENTIAL GROUP RECOMMENDATIONS
Master's thesis
Tampere University
Master's Degree Programme in Information Technology and Communication Sciences
April 2024

In this modern era, sequential recommendations are increasingly prevalent. Users now expect systems to remember past interactions rather than treating each recommendation round as a separate event. Likewise, more and more applications allow users to gather in groups for various activities, such as dining out or movie nights, leading to an increase in the use of group recommendation systems. However, these systems face challenges in processing complex data such as historical user feedback and complex characteristics of the group members. To address these challenges, the SQUIRREL framework is created for implementing Sequential Recommendations with Reinforcement Learning. This framework employs reinforcement learning methodologies to determine the optimal group recommendation algorithm according to the present conditions of the group members. During each round of recommendations, it assesses member satisfaction and item relevance to choose the algorithm that generates the highest reward. While recommending a set of items for a group it is often seen that some members are neglected over others. This makes the recommended list biased to some users' preferences. This work identifies this bias issue and incorporates two new fairness aware reward functions in the original SQUIRREL using the m-proportionality measure. This approach guarantees fairness in recommendations by striving to incorporate a minimum of **m** items in the group recommendation list that align with the preferences of each member. By incorporating these new functions, the SQUIRREL framework demonstrated its adaptability to changes for additional variables. Evaluation results on the MovieLens dataset illustrate the effectiveness of these new reward functions.

Keywords: Sequential recommendations, Group recommendations, Fairness in recommendation systems.

The originality of this thesis has been checked using the Turnitin Originality Check service.

Contents

1 Introduction

In the current digital environment, recommendation systems are essential tools that assist users in finding what they need from a wide range of possibilities. These systems utilize a variety of techniques to forecast human preferences. Their main aim is to make user experiences more personal by suggesting things that match their tastes. Recommendation systems are found in many areas, such as online stores, streaming platforms, social networks, and content websites. By analyzing users' behavior recommendation systems can offer custom suggestions, leading to increased user satisfaction and more successful businesses. To provide efficient suggestions, the system should consider not just the user's present preferences but also their previous experiences with the system. This means considering not just the present session but also past interactions and how the user responded to them, such as which items they chose or liked. The system ought to examine a series of previous interactions or suggestion rounds instead of concentrating only on the present session. This approach represents a newer method in recommendation systems, as most systems typically only look at a user's available ratings without considering their past feedback on recommended items. This concept is often termed as sequential group recommendation, as it involves considering multiple rounds of user recommendations over time [1]. Recommender systems are always evolving, using new technology to provide even better recommendations in real time.

A growing number of people are now interested in group recommendations in addition to sequential ones. Unlike individual recommendations, the focus here is on balancing the preferences of multiple individuals. For instance, imagine a couple of friends planning a movie night aiming to and looking for a best-suited movie for the group. The challenge lies in suggesting a movie that appeals to everyone in the group. Group suggestions are more complex since they must address the interests of all group members, not just one. The method for group recommendations uses a single recommendation system that is used by each group member on their own to get their lists. The group recommendation system then applies an aggregation method to these lists to create a list of suggestions for the entire group[2]. The combination of sequential and group recommendations introduces new challenges and complexities, giving rise to a novel research area known as sequential group recommendations. In this approach, recommendations are made by analyzing the group's past interactions over multiple rounds, leading to a more comprehensive user experience. For instance, consider a group of friends planning a weekend getaway. The recommender system could analyze each friend's travel preferences and past experiences to suggest destinations and activities that fulfill the group's collective interests and ensure an enjoyable trip for everyone.

Group recommendations can be approached in various ways, but determining the most suitable aggregation method for a specific test scenario is not always straightforward. The Sequential Group Recommendations via Reinforcement Learning (SQUIRREL) framework was created in order to address this issue. This framework has similar components to any other reinforcement learning based system: the state, actions, and reward. The state depicts the group's current position, the actions include all of the various group recommendation strategies (or aggregation methods), and the reward specifies the system's primary goal. The system assessed the group's condition throughout each suggestion round, considering factors like individual satisfaction levels of group members, and then made a decision on which action to take. In other words, It chose an appropriate group recommendation approach to apply. This resulted in reinforcement learning being a natural choice for sequential group recommendation systems like SQUIRREL. Subsequently, a group recommendation list was generated based on this decision. Then that list was suggested for the group with the accumulated reward. Afterward, the states were updated to indicate the changes that happened for choosing a particular action. The changes include the overall level of satisfaction of the members for the latest round of recommendations.

Recommending a selection of the best products according to the users' interests is the system's primary goal. However, recommendation systems sometimes struggle to find the most suitable items to recommend. Generally, aggregation methods average the users' individual ratings for an item and assign it to the group. The items are then sorted according to their scores, and the top-k items are recommended. As a result, it can be seen oftentimes that the choices of some of the members are dominated and some of them are neglected by the system. This is due to a number of factors. The "cold start" problem is one such difficulty where the system does not have enough information about a user or set of users to anticipate with enough accuracy. A further contributing element may be the system's ignorance of group dynamics. A group can be formed by different kinds of people with similar and dissimilar interests. This creates biases in the recommended item set for the group, resulting in dissatisfaction for the group. As a result, it is really important to make such a system that can balance the needs of every group member to increase the overall level of satisfaction for the group. Furthermore, the system's configuration which includes thresholds and hyperparameters can have a significant influence on the caliber of the recommendations.

This work extends the SQUIRREL framework for sequential group recommendation to ensure fairness. In the original framework, the states are defined as the current satisfaction level of the users, and 6 different aggregation methods are used as actions. Besides, two reward functions are used for the feedback of the selected action. Here, two new fairness aware reward function is introduced based on the definition of proportionality. Any member of the group must have a minimum number of items on the final group recommendation list in order to meet the definition of proportionality. The list may be referred to be m-proportional if each member has at least m items in it. This value of m-proportionality and its combination with satisfaction is used as two new reward functions. This new reward function aims to tackle the biases that the system can generate. Finally, the performance of these new rewards is evaluated using two different metrics called *groupSatisfaction* and *groupDisagreement*. Also, the quality of the recommended items is measured using Normalized Discounted Cumulative Gain (NDCG) values.

The following sums up this work's total contribution:

- The necessity for fairness metrics in the sequential group recommendation area is addressed in this study.
- It uses the SQUIRREL framework for sequential group recommendation systems and extends it further to incorporate fairness.
- Two new fairness aware rewards are introduced based on the concept of proportionality and its combination with the users' satisfaction.
- Their performance is evaluated using two different metrics called *groupSatisfaction* and *groupDisagreement*. Additionally, Normalized Discounted Cumulative Gain (NDCG) ratings are used to evaluate the efficacy of the recommended items.

The 20M MovieLens dataset is used in all of the studies.

2 Related Works

2.1 Recommendation Systems for Groups

A group recommendation system is a type of recommendation system that aims to suggest items that will be well-received by the majority of a group of people. Unlike traditional recommender systems that focus on personalizing suggestions, group recommendation systems consider the diverse preferences within a group and strive to identify items that align with a shared consensus. In a variety of contexts, it is becoming increasingly pertinent to recommend items to groups of people rather than to individuals, including work-related brainstorming sessions and social activities like family movie nights or friend group outings. However, group recommendation poses unique challenges, particularly due to the fluctuating nature of group compositions and the varying preferences of individuals within each group [3]. As a result, the field of group recommendation has been subjected to thorough scholarly exploration that advances our understanding and methodologies in this area[4].

Through the comprehensive research, two principal methodologies have emerged within the domain of group recommendations [5]. In the first approach, [6], [7] a virtual user is created by aggregating the ratings provided by each group member. This virtual user is then treated as a single entity, enabling the use of standard recommendation methods designed for individual users. The second method, which is quite popular for group recommendations [8], [9] [10], includes using a single-user recommendation system. Then apply this system to each person in the group separately. The different lists of suggestions are then combined to create a single, complete group recommendation list. In my study, I chose the second way since it gives flexibility [11] and opens new opportunities for improving the efficacy of the recommendation process for the group.

A group recommender system takes into account a number of factors in the aggregation phase to improve the recommendation process. For example, the research described in [12] suggests a model to recommend items for groups, that takes into consideration the unique impact of every group member during aggregation. The model emphasizes assigning greater weight to individuals with a higher level of knowledge about the items being considered. Furthermore, using current developments in neural collaborative filtering and attention networks, [13] develops an efficient aggregation technique depending on available data. Within the framework of neural collaborative filtering (NCF), an attention mechanism is employed to alter the group's representation and to grasp the connections between groups and items. This model enhances not only group recommendations but also individual user recommendations, with a particular emphasis on addressing the needs of cold-start users who lack prior individual interactions. Likewise, [14] combines a Bipartite Graph Embedding Model (BGEM) with an attention mechanism to determine how each member affects the group's overall decision-making process. To determine each user's social influence, the attention mechanism was employed, adapting it to different group contexts. To further integrate users' comprehension of local and global social network architecture, they unveiled a novel deep social influence learning framework. With this paradigm, the evaluation of users' social impacts is expected to be significantly improved. In addition, [15] presents a preference-oriented social network technique that uses group member interactions to determine the ultimate group decision. This approach is noteworthy because it makes use of the dynamics of social ties inside the group instead of total access to personal preferences.

Researchers have explored innovative approaches to enhance the aggregation strategy and cater to the dynamic nature of decision-making within groups. One research [16] identified an efficient aggregation technique by modeling the interactions between group members as various voting processes. This was done by attentively studying the interactions between group members. To understand the complex voting strategies used by group members, a stacking social self-attention network was devised, which mimics the process of reaching a consensus. Concurrently, another research [17] addressed the challenge of recommending to large groups by segmenting them into subgroups based on individual interests. In order to aggregate collaborative filtering suggestions, this strategy included first identifying probable candidate media-user couples for each subgroup. In addition, a bimodal group recommender [18] was developed, aiming to satisfy both the group's overall satisfaction and the satisfaction of each individual member. During the first stage, the recommender prioritizes meeting the needs of the whole group; in the second stage, it filters away unnecessary things in order to fulfill individuals preferences.

According to [19], every group member receives a utility score that indicates how relevant the suggested products are to their own preferences. The algorithm then makes an effort to produce an optimal group suggestion list by distributing these utility ratings across the group members. Similarly, a utility definition for users based on the similarity between individual and group suggestions is introduced by [20]. In their process, sets of N-level Pareto optimum items are taken into account when the group recommendation list is being created. [21] proposes the idea of rank-sensitive balancing in the aggregation process, highlighting the need for the first proposal to optimally balance the interests of each group member. This principle extends to subsequent recommendations, ensuring a continual balance in the group's interests with each additional item. Also, [17] present the dynamic group aggregation system (DGAS) as an alternative to traditional aggregation methods. While traditional approaches assign equal weight to each group member, DGAS introduces subgroup weights that consider subgroup contributions and interests, addressing the limitations associated with existing techniques. This innovative approach aims to better reflect individual contributions within the group dynamic.

2.2 Sequential Recommendations

Sequential recommendations involve suggesting items in a step-by-step fashion, considering the chronological order of user interactions or preferences. This methodology is commonly employed in situations where user tastes change over time, as seen in recommending movies, books, or products. Streaming services such as Netflix are a common example, where recommendations are tailored based on a user's viewing history as well as previous preferences. The objective is to anticipate what the user may find appealing next by analyzing their previous selections and the patterns that emerge in their behavior over time.

Typically Session-based recommendations, Session-aware recommendations, and Last-N interactions-based recommendations are the three primary categories of sequential recommendation systems, dependent on the amount of prior interactions they take into account [22]. A user's past N actions are the only ones considered in the Last-N approach [23]–[25]. This is due to the large amount of redundant or unhelpful past data that the system keeps track of for every user. This can overwhelm the system, so it focuses on the most recent actions instead of all the available data. Session-based recommandations approach, on the other hand, are limited to the activities a user takes during their current session and do not take into account any previous interactions. Instead of looking at everything a user has done before, these recommendations specifically pay attention to what the user is doing during their ongoing session. This type of recommendation are commonly used in areas like news and advertising platforms [26], [27]. In contrast, Session-aware recommendation systems has details about the most recent interaction with the user as well as their overall history. With the potential to offer more precise and tailored recommendations, these recommendation systems can improve the user's experience. This improvement is achieved by considering not only the most recent interaction but also taking into account the user's previous actions. Some common applications of these recommendation systems include e-commerce and app recommendations [28]–[30]. A neural network based music recommendation system is introduced in [31]. This system is aware of the users sessions while recommending musics. The neural network architecture used in this application allows the users' preferences to be represented as a series of embeddings. As the system is aware of individual session, it generates one embedding per session. The algorithm determines the best music for a user based on their recent interactions and session-level data (such as the time of day and device activity). In order to assure fairness, [32] proposes a multi-round recommender system that makes use of Variational Autoencoders (VAEs) and introduces randomness into their routine [33]. However, [34]–[36] penalizes ratings given to things based on their past popularity in order to lessen bias and promote diversity.

A paradigm for exploring collaborative knowledge by examining relationships in a bipartite graph is used in the system outlined in [37]. The connections between the nodes in this graph, which depict users and objects as nodes, indicate interactions. Instead of only considering directly connected nodes, the approach also looks at 2-hop neighbors, termed high-order collaborative relations. The system attempts to better capture the dynamics of both user and object behaviors by integrating historical sequences for both users and items. The study presented in [38] presents the GLS-GRL system, which records user-item interactions using two item-item co-occurrence graphs. While one graph concentrates on the present, the other includes all historical data. The GLS-GRL system generates representations for users that take into account their short- and long-term preferences through the application of graph representation learning. Integrated user profiles are created by combining these representations. To improve group representations used in suggestions, the system also uses a user-interactive attention mechanism that takes into account relationships among group members.

This research is build on top of the innovative SQUIRREL model [1]. It is a framework that applies the idea of reinforcement learning to sequential group suggestions. This model incorporates sequential group recommendation techniques initially introduced in [39], [40]. By considering the group as a single unit and dynamically determining a weight based on member satisfaction, the Sequential Dynamic Adaptation Aggregation (SDAA) approach functions. This weight is then used to combine two scores: the item's average preference rating for all group members and the item's preference rating for the user who indicated the least amount of pleasure during the last round of suggestions. In contrast, the Sequential Individual Adaptation Aggregation (SIAA) technique examines each group member separately in order to adopt a tailored approach. In every iteration, it computes an individual weight for each user, taking into account the degree of disagreement stated in the previous round of recommendations as well as their overall satisfaction level. The Avg+ method differs from other methods by considering the entire dataset instead of analyzing each item individually. To capitalize on higher satisfaction levels, it enhances the classical average technique by incorporating more than the specified k items. In the group recommendation list, the algorithm gradually adds items that have the lowest ratings for disagreement in relation to the remainder of the list. In general, the aforementioned research works primarily revolves around recommendation systems designed for individual users.

The practical studies conducted by[41] and [42], explores different ways of combining recommendations for TV episodes and music tracks. These studies were conducted by targeting groups of users in specific situations. In their research, methods like Least Misery, Average Strategy, and Average Without Misery Strategy were employed to select sequences of episodes for group viewing. Additionally, the 'Balancing' approach is employed to propose a sequence of songs that consistently maintain a balance in the user's happiness. It is nonetheless noteworthy that the SQUIRREL model is the first to select an aggregation method from a collection of ways for each round of group recommendations using a machine learning approach known as reinforcement learning.

2.3 Fairness in Group Recommendations

To create an individual preference list of items for each member of the group, we employ a conventional single-user recommendation method when attempting to propose items for a group of users. Later, we aggregate these individual lists into one list for the group. Different aggregation methods can be used to combine multiple lists into one list. These aggregation methods can increase dissatisfaction among the members of that group. As a result, it's critical to pay attention to maintaining fairness while recommending items to a set of consumers. Fairness is a difficult idea that has been interpreted in a variety of ways. However, there is a general consensus that fairness means treating people equally and without prejudice. Fairness in recommender systems refers to the idea that suggestions shouldn't be skewed toward or away from any certain set of individuals [43]. This implies that rather than taking into account the user's ethnicity, gender, age, or any other protected factor, suggestions should be made based on their unique preferences and requirements. Many studies have been carried out recently to include the concept of fairness in group recommendations. However, no formal definition of this concept has been presented so far. Different researchers have presented several new concepts for implementing fairness in group recommendation systems.

In [44], the authors discuss fairness in the context of a problem where they aim to find a ranking that closely matches given sets while adhering to a fairness condition. The emphasis lies in item rankings, where each item is linked to a protected attribute that separates the dataset into many categories. They suggest a fairness criterion to guarantee parity between these groups, focusing on the top-k segment of the rankings, or top-k parity. In simpler terms, they are exploring how to make rankings fair by considering the characteristics of the top-ranked items and ensuring fairness across different groups in the dataset. Besides, in [45], fairness is evaluated by looking at how satisfied each person in a group is with the recommended items. The degree to which each person finds the recommended items relevant determines the level of satisfaction, or usefulness. They determine the average level of satisfaction with a group of items in order to determine the usefulness of that set for a user. Averaging the satisfaction scores of all group members yields the overall satisfaction for a group's recommendation list, also known as social welfare. Then, fairness is evaluated by contrasting the various group members' satisfaction scores. In simple terms, the goal is to create a recommendation list that minimizes dissatisfaction among group members, making it as fair as possible.

In contrast to earlier methods, [20] take into account an item's position in the recomendation list rather than emphasizing the degree to which group members are happy with the recommendations. They use the idea of Pareto optimality, where an item is optimal if no other item ranks higher than it, meaning it's not dominated by any other item. For example, there should be no item j before the item i in the recommandation list. Thus we will say i as Pareto optimal. N-level Pareto optimal extends this concept to identify the N best items for recommendation, ensuring fairness by including each user's top choices. Additionally, [21] introduces the concept of rank-sensitive balance to help in combining individual preferences. The first recommendation aims to balance all members' interests, and this balancing act continues with each subsequent item in the list. In simpler terms, these approaches aim to create fair group recommendations by considering the ranking positions of items and ensuring each member's top choices are included.

A method for evaluating fairness makes use of voting and game theory. Conflicts of interest between members of different random groups are handled in [46] by using non-cooperative game theory [47]. This method helps resolve disagreements among members in different groups by using a specific type of game theory. [48] assumed that the recommender system has a good understanding of how users typically rate items. Using voting theory, the system suggests a ”winning” item to the group based on this probabilistic knowledge of user preferences. In short, this recommendation system uses what it knows about how users usually rate things to suggest a preferred item for the group.A innovative technique of group recommendation is proposed in [49], wherein each member of the group is able to express their ideas on the decisions made by the rest of the group. This enables users to receive new suggestions that align with what others in the group have proposed and also allows them to explain the reasons behind their own alternative suggestions. [8] introduces a different way to ensure fairness in group recommendations. They propose a consensus function that considers the group's opinion using the average method and also takes into account the differences in opinions among users. This divergence of views is expressed as the variance disagreement function, which computes the mathematical variance in the relevance ratings provided by group members for the item, or as the average difference in how users assess an item.

Recently, in [50], two fairness definitions were introduced. According to the first principle, known as fairness proportionality, a user deems a recommended list to be fair if they find at least a particular number of items appealing. Let's say a group recommendation list has k number of items in it. In order to be proportional, this list must have a m number of items from each user's individual recommendation list. The second definition is envy-freeness, when the recommendation list includes a minimum number of items from a user that don't make the other user feel envious, then they see the list as fair. This research draws inspiration from these fairness definitions and uses a similar approach, incorporating users' satisfaction and disagreement scores. Additionally, in this work, proportionality fairness measures are directly used as a reward function in the SQUIRREL model.

2.4 Recommendations through Reinforcement Learning

In recent years, there's been a shift in how we approach recommendation problems. Instead of viewing it as a simple task, it's now seen as a step-by-step decision-making problem where each decision is based on the outcomes of previous ones.This is kind of like a game where the best moves depend on the current situation. One can represent this using Markov decision process (MDP). The exciting part is that we can use reinforcement learning (RL) techniques, especially deep reinforcement learning (DRL), to tackle recommendation challenges that involve massive amounts of actions and states. This has given rise to what we call RL-based recommender systems (RLRSs), which essentially use the power of DRL to make smarter and more effective recommendations[51].

An initial contribution to this field was made by [52]. The last N sites the user had viewed served as the system's environment when they developed a recommender system for the web. The suggested page's ranking and the amount of time the user spent on it were combined to create the reward system, and the actions consisted of recommending pages. A Deep Q-Networks (DQN)-based reinforcement learning framework for personalized online news recommendation is presented in [53].In order to deliver more diversified and effective suggestions in the online news realm, this technique seeks to provide a thorough understanding of user preferences and news dynamics. There are two primary stages to the framework's structure: offline and online. During the offline process, the model undergoes training, while in the online phase, the agent generates recommendations and records user feedback. Following a certain period, the model then moves back into the offline phase, where it may be modified in response to the input that was recorded. The system effectively captures dynamic news features and user preferences to enhance long-term rewards. Besides click/no-click feedback, the authors also take into account user return patterns to perceive user behavior. Also, they incorporate an exploration strategy to promote diversity in the recommendation.

In the study by [54], reinforcement learning methodologies are utilized to enhance the recommendation model's long-term accuracy. This system demonstrates a significant improvement compared to previous methods. The work particularly addresses two key aspects of recommendation systems: the cold-start and warm-start scenarios. Using the recommender system as the environment and users as agents, this method takes advantage of their interactions. The model's flexibility makes it useful in cases when there is not enough data, such as when there is little content information. In the work presented in [55], a pioneering recommender system is proposed, using Reinforcement Learning (RL) to continuously refine its methods through interactions with users. As a Markov Decision Process (MDP), the authors understand these interactions. Before deploying the model online, they make it easier to pretrain and evaluate its parameters offline. Then they introduce an online interactive user-agent environment simulation system. In the course of user-agent interactions, the research highlights the importance of recommendations that are provided list-wise. In order to smoothly include these list-wise recommendations into the recommendation architecture, it presents a novel method called LIRD. This study tackles the shortcomings of current recommender systems, which frequently give precedence to instantaneous satisfaction over lasting benefit. By employing RL, the study aims to identify recommendation methods that optimize long-term advantages.

To monitor the evolution of user preferences, [56] proposes an innovative movie recommender system incorporating deep learning, reinforcement learning, and prioritized experience replay. The utilization of reinforcement learning involves employing agents to grasp user preferences and movie characteristics. The model is able to produce suggestions based on changing choices because it incorporates prioritized experience replay, which helps it to capture changing user interests. By simplifying the intricate process of capturing relationships and patterns between viewers and films, the deep learning technique used produces recommendations that are more precise and unique. Identifying and adjusting to users' changing preferences over time is a problem that this method offers a fresh and useful answer to.

The research presented in [57] introduces a user-centric framework for sequential music recommendation.The main focus was on a scenario where users have access to content from various channels. In this context, a personalized recommender system was implemented on the user side, determining the most suitable channel based on users' listening patterns. Explicit and implicit user feedback is integrated by this technology into a Markov Decision Problem (MDP). This improvement stemmed from the integration of reinforcement learning techniques, facilitating the system's decision-making process for the next channel to play. Explicit input was derived from users' preferred music channels, while implicit feedback was gathered through their requests for new songs. In [58], an MDP-based recommender is used for a commercial system, and they suggest representing the current state of the environment with an ordered list of choices made by each user. Based on this user behavior, their algorithm seeks to foresee and suggest new items. They point out that theirs is one of the few commercially available recommender systems, which is noteworthy. It also represents the first disclosure of experimental research results from a real commercial location, showcasing the business-useful application of their MDP-based technique.

All of the research shown so far have used reinforcement learning to solve the recommendation issue, although they generally focus on a single recommendation domain. On the other hand, the SQUIRREL framework [1] aims to be versatile across various domains and has the capability to blend different aggregation techniques. This flexibility is designed to address the inherent limitations present in existing recommendation solutions. Essentially, this approach strives to offer a more adaptable and comprehensive solution to the diverse area of recommendation problems.

3 SQUIRREL Model for Sequential Group Recommendation

Recommender systems are crucial in modern digital platforms, helping users find various items such as products, movies, and music. They have undergone notable advancements, progressing from traditional collaborative filtering and content-based methods to state-of-the-art methods like reinforcement learning. Reinforcement learning (RL) represents a machine learning framework where an agent learns decision-making skills through engagement with an environment. The agent perceives the present state of the environment, executes actions, and gains feedback through rewards. The objective is for the agent to acquire a strategy that optimizes the total reward accumulated over time. Every reinforcement learning framework has the following components.

- **Agent:** The decision-maker, or learner, interacts with it's environment.
- **Environment:** An external system that interacts with the agent, providing feedback.
- **Actions:** Choices or moves available to the agent within a given state.
- **Rewards:** Feedback indicating the quality of the agent's actions from the environment.
- **Policy:** The strategy or behavior guiding the agent's action selection.

This structure of reinforcement learning can be effectively used in the domain of sequential recommendation problems. Both involve making decisions across a series of steps. In reinforcement learning, actions are chosen sequentially to optimize long-term rewards, whereas in the sequential recommendation, items or actions are suggested in a sequence to enhance user satisfaction or engagement. There is a feedback loop for both instances which helps to refine decision-making or recommendation strategies over time. In other words, we can say that both techniques have a similar nature to optimize their goal in a dynamic environment. In reinforcement learning, the environment could change by the action of an agent or external factors. On the other hand, in sequential recommendations, changes are due to users' preferences or the availability of the items. With these similarities in consideration, RL is a natural choice for any sequential recommendation problem. Consequently, a framework known as SeQUentIal Group Recommendations via ReinforcEment Learning (SQUIRREL) [1] was created.

3.1 The SQUIRREL Framework

To understand the SQUIRREL framework properly, we need to understand how Reinforcement Learning works in Group Recommendation systems. Assume that U represents a collection of users and I represents a set of items to recommend. With $G \subseteq U$, these users can create a group G. Every member u of a group has their own ordered list of preferred items. Generally, this kind of list is created using the single recommendation technique. We can denote such lists for a user u in a specific recommendation round j as B_u^j. The member lists are then combined to create a single list of recomended items GL_G^j for the group as a whole. To create a single, distinct list, the agent in RL choose a course of action based on the group's present situation. Based on the action chosen by the agent, it receives a positive or negative reward which helps the agent to make a perfect policy for future recommendation rounds. This configuration is similar to a Markov Decision Process (MDP). Any MDP can be represented using the following tuple (S, A, P, R):

- **S** is a limited collection of states. It represents the condition of an environment at different times.
- **A** represents a limited collection of actions. These are the available options for an agent to perform on the environment.
- **Pa(s, s')** denotes the transition probabilities between states. This reflects the probability of transitioning from one environmental state to another environmental state after completing a certain activity.
- **Ra(s, s')** indicates the benefit obtained upon transitioning from environmental state s to s'. Based on the status of the environment at the time, the reward function's result represents the quality of a selected action. It assists in figuring out the agent's best course of action.

Similar to this, an agent interacts with environment E in the SQUIRREL model in order to maximize the overall reward accrued following each recommendation round. SQUIRREL model uses users utility scores as state, different aggregation techniques as action, and group utility scores as rewards. As a result, this model creates a μ number of recommendation lists $GR = (GL_G^1, GL_G^2, \ldots, GL_G^{\mu})$ for a group in different rounds of recommendation.

A SQUIRREL model recommendation round's structure is shown in Figure 3.1. Each group member's individual list, represented as B_u^j, is sent to the group at the beginning of each round j using a single-user recommender system. After that, the SQUIRREL model processes these lists. The model assesses the current environmental state, primarily focusing on the group's satisfaction level S_j. Based on this

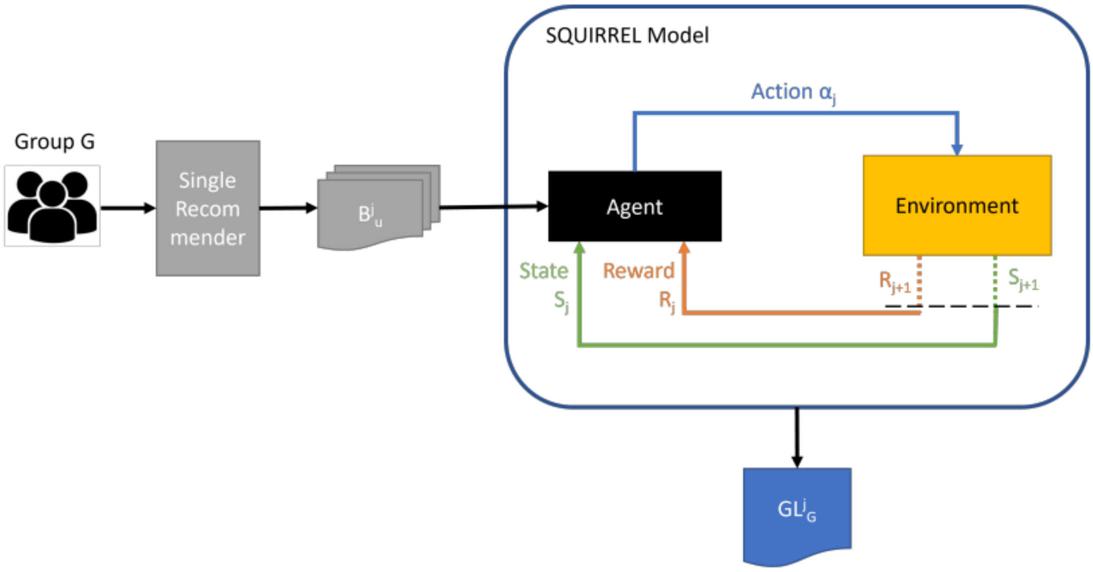


Figure 3.1 *The SQUIRREL Model*

observation, the model chooses a suitable action α_j and combine all the individual recommendation lists of the group members. Following this action, the user's overall satisfaction and the computed reward R_{j+1} are updated, resulting in a transition to the following state S_{j+1}. The resulting group suggestion list GL_G^j is the last thing the model gives the group. To ease the understanding, Table 3.1 provides explanations for all the symbols and notations used in the SQUIRREL model.

Notation	**Definition**	**Notation**	**Definition**
U	Set of users	GR	Sequence of group recommendations
I	Set of items	GL_G^j	Group recommendation list at round j
u	User	SQUIRREL Model	
G	Group	μ	Number of rounds in the sequence
d	Item	S	State of the environment
j	Round	A	Set of actions
B_u^j	Recommendation list for user u at round j	$P_a(s, s')$	Probability of transitioning from state s to state s'
$B_{u,k}^j$	Top k recommendations for user u at round j	$R_a(s, s')$	Reward from transitioning from s to s' under action a
$p_j(u, d)$	User u's relevance score for item d at round j	π	Policy of the model

Table 3.1 *Notation and Definitions used in the SQUIRREL [1] model*

3.1.1 Definition of state in SQUIRREL Framework

The definition of the environment's state is a vital component of the SQUIRREL model. This state reflects the current situation of each group member. It's important to have a state that specifically addresses the needs of each member to ensure the model can make optimal decisions without overlooking anyone. Focusing solely on the group as a whole might obscure individual requirements, causing the decision-making process to possibly make mistakes.

Within the SQUIRREL framework, two user utility functions have been used to define the states. One is their satisfaction with a recommended item and another is their disagreement with the other users of the group. These two utility scores give an idea to the agent to understand the situation of the individual user of the group. To calculate user satisfaction this model uses their list of preferences B_u^j in a recommendation round j generated from the single recommender system. Then, it compares the best items let's say top k items of that list to the recommended list for the entire group.

Assume $B_{u,k}^j$ is a list containing the top k items with the highest scores in B_u^j, and GL_G^j is the group recommendation list likewise containing k items. In order to determine a user's level of satisfaction with the items in the group suggestion list, we will compare them directly to the items in $B_{u,k}^j$, which, for the user, are the best choices. In terms of group recommendations, this contrast is typically referred to as the individual loss.

$$\mathrm{sat}(u, \mathrm{GL}_G^j) = \frac{\sum_{d \in \mathrm{GL}_G^j} p_j(u,d)}{\sum_{d \in B_{u,k}^j} p_j(u,d)} \tag{3.1}$$

Here, $p_j(u,d)$ is the score of an item d in a list. The total of the scores for the top k items in the group suggestion list is determined by the numerator of equation 3.1. Conversely, the denominator provides the total of the ratings for the first k items in each individual list. After each round of recommendations, this new set of normalized satisfaction scores for every user is treated as the new state of the environment.

3.1.2 Actions in SQUIRREL Framework

The actions serve as the key drivers within the model. The agent evaluates the environment's state and its past reward history, then chooses an action to apply. These actions make changes in the state of the environment, which subsequently leads to the computation of a reward. Since actions are the sole means of introducing change in the environment, they are the foundation of the group recommendation process by nature. As stated earlier, each group member receives a unique list generated by a single recommender system, which is regarded as a personalized tool in the model. As a result, in order to merge these various lists into a single group recommendation list, the group recommendation process employs many strategies, each of which stands for a particular action. Six distinct aggregation functions are employed in the SQUIRREL model as actions to merge the several lists into a single, condensed list of items for the group.

- **Average:** This technique is aggregating based on the conventional average methodology, in which the average of all the expected scores that each group member has assigned to a particular item determines the group's anticipated score for that item. In this case, each group member's anticipated score for an item is equally important.

- **RP80 [8]:** Group disagreement scores and group relevance are the two components that this technique combines. For item d, group relevance is *avg(d, G)*, which is the average predicted score for every member of group G produced by a single recommendation system. For item d, the average pairwise disagreement between the anticipated scores provided by the group members is the group disagreement, denoted as *dis(d, G)*. It measures how well group members agree on the item using relevance ratings. Within group G, the ultimate score for item d is determined as follows:

$$RP80(d,G) = (1 - w) \times avg(d,G) + w \times (1 - dis(d,G)) \tag{3.2}$$

  Here, w acts as a tuning parameter that modifies the relative importance of disagreement and group relevance.

- **PAR [45]:** Using an iterative selection process, this method creates a group recommendation list by finding things that best balance the ratings for Social Welfare (SW) and Fairness (F). Social Welfare reflects the average satisfaction of all group members with a particular item, a metric similar to the satisfaction measure of the model. Fairness, denoted as *F(d,G)*, is calculated as the variance in satisfaction scores among group members. The overall score of an item for the group is determined by the following equation

$$PAR(d,G) = \lambda \times SW(d,\ G) + (1 - \lambda) \times F(d,\ G) \tag{3.3}$$

  Here, λ is a parameter that regulates the weightage between Social Welfare and Fairness

- **Sequential Dynamic Adaptation Aggregation (SDAA) [40]:** Weight w_j is dynamically calculated at round j in the SDAA technique. The difference in satisfaction scores between the most and least happy group members from the previous recommendation round determines this weight. To calculate w_j for SDAA, one must first take the highest satisfaction score of group members at round *j–1* and subtract the minimum satisfaction score of group members at round *j–1*. With the help of this weight, the projected score of the least pleased member is balanced with the average expected score of an item for all group members, which is represented as $avgG(d, G, j)$. Round j yields the following final score for item d for group G:

$$SDAA(G, d, j) = (1 - w^{j}_{SDAA}) \times \textit{avgG(G,d,j)} + w^{j}_{SDAA} \times \textit{leastG(d,G,j)} \quad (3.4)$$

- **SIAA [40]:** This technique takes into account both the customer satisfaction and disagreement scores. A user's disagreement score, u, also known as *userDis*, calculates the discrepancy between their level of satisfaction and the highest-scoring member. SIAA conducts individual analyses of every group member. It determines a weight for each member by taking into account their total satisfaction and disagreement from the previous round of recommendations. Using a parameter b, this weight, represented as $w^{j}_{SIAA}(u)$, balances user disagreement and overall satisfaction scores.GR^{j-1} denotes the suggestions from all earlier iterations in the j^{th} round of recommendations. This weight is immediately assigned to each item's anticipated score when the single suggestion procedure is finished. The average of each group member's weighted score for an item then becomes the final prediction score for that item. The equation for $w^{j}_{SIAA}(u)$ is as follows:

  $$w^{j}_{SIAA}(u) = (1 - b) \times (1 - satO(u, GR^{j-1})) + b \times userDis(u, GL^{j-1}_{G}) \quad (3.5)$$

  In this case, user disagreement ratings and total satisfaction are balanced using b as the weight.

- **AVG+ [40]:** Two stages make up the Avg+ aggregation technique. The average aggregation is used first. The second phase involves iteratively adding items to the group recommendation list that reduce the group disagreement score (*groupDis*). Group disagreement may be defined as the variation in satisfaction ratings between the group's most and least happy members. Following this, GL^{j}_{G} is modified in the second phase:

  $$GL^{j}_{G} = GL^{j}_{G} \cup \{d | min_{\forall d \in AvgList^{j}_{G}}(groupDis(GL^{j}_{G} \cup d)) \vee d \notin GL^{j}_{G}\} \quad (3.6)$$

In this case, the list of the top k items following the first round of aggregation is represented by $AvgList_G^j$. The probability of changing from state s to state s' in round j by taking action and is defined by $P_a(s, s') = P_r(s_{j+1} = s' s_j = s, a_j = a)$.

Lastly, the reward earned while changing from state s to state s' is represented by $R_a(s, s')$. This reward signifies the effectiveness of the recommendations made by the model. By evaluating the reward received from taking an action, the model determines whether the action is suitable or not. The definition of the reward can be customized according to the objectives of the model. Generally speaking, we want the model to recommend items that are pertinent to the whole group. In Section 4, the rewards used in the SQUIRREL model are explained in more detail.

4 Reward functions in SQUIRREL Framework

The reward obtained from an action, considering the environment's state, serves as the sole criterion for the model to assess the appropriateness of its actions. This reward can be highly adaptable in its definition, depending on the objectives of the model. Typically, we aim for the model to suggest items that recommend the best possible items for the entire group. Determining group utility scores which express how well the recommendations work for the entire group is essential to achieving this. The SQUIRREL model establishes diverse reward functions that take into account all group members collectively, rather than focusing on individual members separately.

4.1 Satisfaction and Disagreement Rewards

In SQUIRREL the primary approach to defining a reward function is through the group satisfaction score. Here, group satisfaction is represented by extending the idea of individual user satisfaction level. The system's ability to fairly balance each member of the group's unique preferences is shown by this score. A high group satisfaction score implies that the system finds items that are important to most group members, whereas a low score shows that the system is not doing a good job of identifying relevant items.

A Group G's satisfaction score for a group recommendation list GL_G^j is determined by averaging the satisfaction scores of each member of the group.

$$\text{groupSat}(GL_G^j) = \frac{\sum_{u \in G} \text{sat}(u, GL_G^j)}{|G|} \tag{4.1}$$

After that, it is possible to compute group G's overall satisfaction across a recommendation sequence GR made up of μ numbers of group recommendations. The average happiness score for GR among the group's users is what this total group satisfaction is defined as in the following equation.

$$\text{groupSatO}(GR) = \frac{\sum_{u \in G} \text{satO}(u, GR)}{|G|} \tag{4.2}$$

This total group satisfaction measure is then employed to indicate the reward received as a result of action a in the recommendation round j. The following illustrates the reward function that is based on the satisfaction criteria.

$$R_s(GR^j) = \text{groupSatO}(GR^j) \tag{4.3}$$

Here GR^j indicates all the recommendation rounds from the starting to j^{th} round.

Considering the overall group satisfaction as a reward function can be a disadvantage due to the nature of its calculation metrics. Here, the SQUIRREL model takes the individual users' satisfaction scores in each round and takes their average to aggregate them for a group recommendation list. This average metrics can cause problems. Let's say for a group of users, all of them have higher satisfaction scores except one or two. The overall group satisfaction will still be high enough and those poorly satisfied users may be ignored. This indicates that reward only based on satisfaction is not enough to take care of every member of the group. As a result, another reward function considering the disagreement of the users is added in the SQUIRREL model. The difference between an individual's satisfaction score and the highest satisfaction score achieved by all other group members is known as the disagreement score of that user. This approach helps us identify whether a group member is consistently receiving preferential treatment (shown by extremely low user disagreement ratings) or being neglected (shown by extremely high user disagreement ratings).

$$\text{userDis}(u, GL_G^j) = max_{\forall u_l \in G}\text{sat}(u_l, GL_G^j) - \text{sat}(u, GL_G^j) \tag{4.4}$$

By looking at the lowest and greatest satisfaction levels inside the group at each recommendation round, we may expand on this idea to get a group score.

$$\text{groupDis}(GL_j^G) = max_{u \in G}\text{sat}(u, GL_G^j) - min_{u \in G}\text{sat}(u, GL_G^j) \tag{4.5}$$

Thus, it is possible to calculate the total disagreement of the group G over a recommendation sequence GR that consists of μ group recommendations.

$$\text{groupDisO}(GL_j^G) = max_{u \in G}\text{satO}(u, GL_G^j) - min_{u \in G}\text{satO}(u, GL_G^j) \tag{4.6}$$

To take the best of both the rewards mentioned above, the SQUIRREL model created another reward based on those two. This strategy comprises two components: *groupSatO* and *groupDisO*, representing the preference level of group members towards an item and the extent of agreement or disagreement among group members, respectively. The aim is to achieve a suitable balance between these components.

$$R_{sd}(GR^j) = 2 \times \frac{groupSatO(GR^j) \times (1 - groupDisO(GR^j))}{groupSatO(GR^j) + (1 - groupDisO(GR^j))} \tag{4.7}$$

The ideas of harmonic mean are applied to *groupSatO* and *groupDisO* in this instance. This harmonic mean is commonly referred to as the *F-score*. We use 1-*groupDisO* to represent the group agreement, accounting for the input functions needed for the *F-score*.

4.2 Fairness Reward

Two different reward functions are used in the original SQUIRREL model: R_s, which depends just on the group's overall satisfaction, and R_{sd}, which takes disagreement into account as well. However, the reward function is quite flexible and may be tailored to meet the particular goals of the framework. As a result, this work has introduced a new reward function based on Fairness. The reason behind choosing fairness as a reward is it plays a vital role in recommendation systems to ensure that everyone is treated fairly. Without fairness, some people may be treated unfairly, leading to unequal access to opportunities or resources. Fairness helps prevent unfair treatment or bias based on factors like race, gender, or age. By ensuring fairness, recommendation systems can build trust and satisfaction among users. Fairness also helps maintain the credibility and ethical standards of the recommendation system, reflecting societal values of equality and fairness. Overall, fairness is essential for creating inclusive and diverse recommendation systems that benefit everyone.

In order to guarantee equity while creating a group recommendation list GL_G^j that meets the needs of every group member, this new reward adopts a fairness criterion known as proportionality [59]. This concept is inspired by the idea of fairly dividing resources among individuals. Proportionality aims to consider each group member's preferences by counting their preferred items in the group recommendation list. A user is deemed to like an item if it ranks within the top $\Delta\%$ of their individual recommendation list B_u^j. A group recommendation list is called m-proportional when it contains at least m preferred items for every user. When $m = 1$, it is termed *single-proportional*, while for $m > 1$, it is referred to as *multi-proportional*. The m-proportional approach fosters mutual acceptance of the recommended items among group members. When a list contains at least m items that a user strongly prefers, they are likely to be more accepting of other items in the list, recognizing that other group members may prefer those items. Thus, m-proportionality promotes acceptance and mutual understanding within the group. The definition of m-proportionality can be expressed as follows.

- **[m-PROPORTIONALITY]** For a given group G and its group recommendation list GL_G^j, m-proportionality is calculated as:

$$R_{mprop}(G, GL_G^j) = \frac{|G_p|}{|G|} \tag{4.8}$$

Specifically, G_p denotes the subset of users in G for whom GL_G^j is m-proportional.

Another reward function, R_{smprp}, is introduced in this work which combines the overall group satisfaction with m-proportionality. Mathematically, it is expressed as:

$$R_{smprop}(G, GL_G^j) = 2\frac{groupSatO(G, GL_G^j) * R_{mprop}(G, GL_G^j)}{groupSatO(G, GL_G^j) + R_{mprop}(G, GL_G^j)} \tag{4.9}$$

This function aims to balance the satisfaction of the group and the fairness using the overall satisfaction score and m-proportionality. The objective is to ensure that each member receives some of their preferred items while maintaining the group's overall satisfaction level.

5 Evaluation of SQUIRREL with new Fairness Measures

5.1 Experimental Setup

This section will elaborate in detail on the different settings used during the experiment with the SQUIRREL model. Also, the performance of the new reward function will be analyzed and compared with the existing rewards.

5.1.1 Dataset

The first step of any machine learning application is to gather data that is perfectly aligned with the scope of the problem we are trying to solve. While working and experimenting with the SQUIRREL model, there was no suitable dataset available according to the need of the model[1]. The authors were looking for a dataset that focused on interactions between groups and systems. This dataset needed to contain scenarios where entire groups collectively rated an item. Instead of finding a dataset with group interactions, they made their own groups using data from the 20M MovieLens Dataset [60]. In this work the same dataset is used for the experiment and the performance evaluation.

The MovieLens dataset is made of 20 million ratings for 27,300 movies. These ratings are provided by 138,500 users spanning from January 1995 to March 2015. In order to assess the model's performance over multiple recommendation rounds, it is important to simulate the passage of time. This involves gradually introducing data to the recommender system instead of providing it with all the information at once. To achieve this, the entire dataset is organized based on the chronological order of ratings. Then, after each round of recommendations, add a new chunk to the system by dividing the datasets into several chunks.

The dataset is initially split equally in half. The first section functions as the system's initial dataset and contains the oldest ratings, which were submitted between January 1994 and December 2003. This first dataset includes 6,382 movies and 8,381,255 ratings from 73,519 individuals. The primary motivation for beginning with such a large dataset is to prevent the cold start issue. This problem occurs in recommender systems when a user has insufficient information, which makes it difficult for the system to provide meaningful suggestions. But solving this issue is outside the purview of this study. The timestamps of the remaining dataset are used to break it into smaller chunks. It is split up into 22 chunks, each lasting six months. Of these chunks, only the first 14 are used in the trials. The tests include dividing the dataset equally according to its size. The first part makes up 40% of the whole dataset, while the remaining 60% is split up into 14 smaller parts. Nonetheless, sparsity—defined as the proportion of missing ratings to all potential ratings—is evident in the datasets. Because new users and relatively low-rated things are added to the system with each chunk, the sparsity rises.

5.1.2 Group Formation

The main goal was to assess the model using realistic situations. Since the dataset lacks specific information, such as who are friends or who have similar preferences, an assumption was made that individuals with shared interests would rate items similarly. To achieve the goal a simulation of two distinct real-life scenarios is created. Firstly, when an existing group allows a new member to join the group, their interests may not align with the group's interests. For example, imagine a newcomer joining a movie night group. Secondly, imagine a situation where a group of people at random get together for a book club or other activity. Even though each member has different tastes, they all need to participate in the same activity, like reading a book together.

The model's evaluation of these two kinds of groupings takes into account the members' different degrees of similarity. For the calculation of similarity scores and the formation of groups, the original datasets from the splitting procedure are used in the study. Values between -1 and 1 are calculated by using the Pearson Correlation [61] similarity function. Positive numbers imply user similarity, whereas negative ones reflect more dissimilarity between users.

$$s(u_i, u_l) = \frac{\sum_{d \in X} (r(u_i, d) - \bar{r}_{u_i})(r(u_l, d) - \bar{r}_{u_l})}{\sqrt{\sum_{d \in X} (r(u_i, d) - \bar{r}_{u_i})^2} \sqrt{\sum_{d \in X} (r(u_l, d) - \bar{r}_{u_l})^2}} \tag{5.1}$$

Items rated by both users are included in the set X; they are indicated by the notation $r(u, d)$, where u denotes the user and d is the item. The average rating for user u is denoted by $\bar{r}_u$. Here, two users are defined as similar if their similarity score is 0.5 or higher, and dissimilar if it's -0.5 or lower. During the experiment, the following two types of group structures of 5 members are used to achieve realistic situations.

- **4 similar users and 1 different user (4+1):** In this group setting, the group has four members who are similar to each other. On the other hand, the lone member is different from all other members in the group.
- **5 different users(5 Diss):** Each member of the group has shown different interests to all other members.

5.1.3 Single User Recommendations

During the evaluation process, in each round, the recommendation lists of each group member are combined to create a recommendation list for the group. This is the initial phase of the model, known as a single-user recommendation. It is treated as a pre-calculated knowledge for the model, meaning that the group members' personal recommendation list indicates the true preferences of items for that user. The system doesn't have access to any additional information about the users beyond these lists. Any existing system that can provide a user with a list of recommendations can be used as a single-user recommendation system. Although systems that only create one item can be employed, in these situations the aggregation procedure becomes simple.

The individual recommendation lists of the users in the SQUIRREL model are generated via a user-based Collaborative Filtering (CF) recommender system [62]. This system generates recommendations by identifying users who have similar preferences and then making suggestions for items based on their likes and dislikes. With the Pearson Correlation function, the similarity between the users is determined. A user is deemed similar if they have rated at least five common items and their similarity score is higher than a certain threshold. This threshold can be different due to the nature of the dataset. For the 20M MovieLens dataset 0.8 has been chosen as a threshold for the similarity. The reason behind this high threshold is the sparsity present in the dataset. A Weighted Sum of others' ratings is used to determine a user's preference for an item [63]. According to their similarity scores, the ratings of users who are similar to one another are taken into account when computing a preference score.

$$p(u,d) = \bar{r}_u + \frac{\sum_{u' \in (P_u \cap U(d))} s(u,u')(r(u',d) - \bar{r}_{u'})}{\sum_{u' \in (P_u \cap U(d))} |s(u,u')|} \tag{5.2}$$

A user may have several users' who are similar to them based on their similarity ratings. But for every user, P_u indicates the top 100 most similar users, and $U(d)$ is the sum of all the users who have rated item d.

5.2 Result Analysis

5.2.1 Preface

During the experiment, in every recommendation round, the group got a list of ten recommended items. The group is already familiar with the items recommended in previous rounds, so those items are avoided recommending them again. As we know the experiment considers two types of group settings, one with 4 similar members and 1 dissimilar member, and another with 5 dissimilar members. There are 100 distinct groups produced for each group setting, each with five members chosen from the MovieLens dataset. All these groups are split into 80 and 20 percent. Among them, 80 of them are used to train the model and 20 for testing. During testing, various aggregation functions are evaluated individually, including Average, Par, Avg+, SIAA, SDAA, and, RP80.

The versatility of the model may be limited if it is trained only for one sort of group, but it could be beneficial for specific applications. For instance, a system that specializes in accommodating diverse preferences among dissimilar group members could be advantageous for activities involving randomly assigned groups. However, the aim for the model to be versatile enough for broader applications. In order to do this, 10 groups from the test sets and 40 groups from the training sets were chosen at random for each type of group. As a result, another set of training and testing data is created. It is to be noted that a randomized sequence of groups is chosen during the training phase to prevent the model from being biased to a specific group type.

To implement the SQUIRREL RL model, python is used as a programming language due to its strong community and library support for Machine Learning. The learning policy is based on the Proximal Policy Optimization (PPO) approach and the Tensorforce library. The value γ, which establishes the significance of future rewards, is set to 0.99 in this case, whereas the chance of changing states under a random action is set at 0.3. Since the dataset shows sparsity significantly in its characteristics the optimal learning rate for the dataset needs to be determined carefully. This parameter indicates how quickly the model adapts to the data. After conducting extensive experiments, the best learning rate for the MovieLens dataset is found to be 0.0022.

Additionally, each aggregation method requires some fine-tuning of specific parameters. For instance, in the RP80 aggregation method, the parameter w is set to 0.8 based on previous experiments demonstrated in [8] on the MovieLens dataset, which is also utilized. This parameter plays a crucial role in balancing group relevance and disagreement. Similarly[64], the Par approach combines social welfare and fairness scores using a value of $\lambda = 0.8$. Using the same datasets as in previous empirical study [40], we find that the ideal value for the parameter b in the SIAA aggregation function, responsible for combining user satisfaction and disagreement, is 0.2. Moreover, through experimentation, the number of items with the greatest prediction score, denoted by k, is determined to be 50, yields the best results for the Avg+ method. Besides, the parameters of the R_s and R_{sd} functions remain unchanged as per the original SQUIREL model. In this work, for R_{mprop} and R_{smprop}, we opted for $m = 2$ and $\Delta = 20\%$. These specific values for both m and Δ have been determined to be the most effective based on the outcomes of a separate experiment. The experimental results will be discussed in section 5.3.

It is worth mentioning that similar to other reinforcement learning models, SQUIRREL undergoes a training phase where it learns from a substantial amount of data. During this phase, the model grasps which actions, such as aggregation methods, are most effective given the current situation. If any part of the model, such as its environment or the actions changes, the model must undergo re-training to adapt to these modifications.

5.2.2 Evaluation Criteria

In each round of recommendation, a list of 10 items is suggested by the model to the group. Then, an assessment of the model's performance is required. Also, unique evaluation measures are needed for the evaluation which reflect the true potential of the model. Evaluation metrics will help to understand how well this model has achieved its objective in that particular round. Thus accumulation of those scores of individual rounds will give us a true picture of the model in all the recommendation rounds. As a result, two metrics have been used in the SQUIRREL model named group satisfaction *groupSatO(GR)* and group satisfaction *groupDisO(GR)*. After each round, the average group satisfaction *groupSatO(GR)* and group disagreement *groupDisO(GR)* are computed for each group in the test set. The computation of the reward functions R_s and R_{sd} is aided by these measures. To further evaluate the quality of the suggested items, the Normalized Discounted Cumulative Gain (NDCG) is computed as a supplementary metric. NDCG helps us determine how well the most desirable items for a user are ranked in the group recommendations.

$$\mathrm{NDCG}(u, G, j) = \frac{DCG(u, G, j)}{IDCG(u, G)}, \tag{5.3}$$

$$\mathrm{DCG}(u, G, j) = \sum_{d \in GL_G^j} \frac{|d \cap B_j^u|}{\log_2(r(d, GL_G^j) + 1)}, \tag{5.4}$$

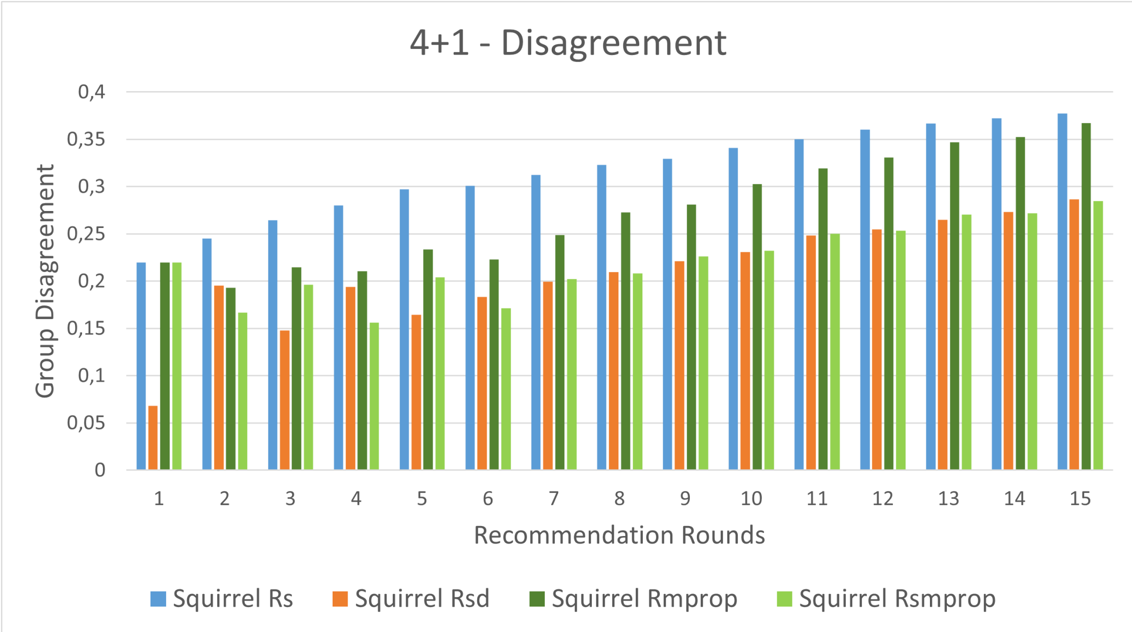
In this case, $r(d, GL_G^j)$ denotes the item d's position in the group recommendation list GL_G^j, and Ideal Discounted Cumulative Gain (IDCG) values are used to potentially indicate the best items for a user u_i.In other words, *IDCG(u,G)* represents the greatest feasible DCG value for the specified user and group G.

5.2.3 Assessing the Satisfaction and Disagreement values

In every recommendation round SQUIRREL model observes the overall condition of the group members. After that, it chooses an action from the list of acts to produce a list of ten suggestions. Based on the items chosen in the current round it receives some rewards and according to the feedback on that reward, it adjusts its future choice of action. Here, four reward functions will be used to test the model's performance. Two of them are from the original model settings and the rest are new reward functions based on the fairness criterion. This model is tested for fifteen rounds, and its performance is evaluated using overall group satisfaction and disagreement scores.

Figure 5.1 shows the group satisfaction and disagreement values for 4+1 group settings. When analyzing the results for the 4+1 group settings across the four reward functions, it is clearly seen that the R_s reward function consistently achieves the highest group satisfaction scores (Figure 5.1(a)) in all the recommendation rounds. Conversely, for the R_{mprop} function this model tends to have lower satisfaction scores except in the initial round. This discrepancy arises because R_s aims to maximize overall group satisfaction, while R_{mprop} focuses on recommending a fixed number of preferred items per user. In the current setup of the model, satisfying every group member becomes challenging, especially since at least one member often has preferences that differ significantly from the rest. Besides, R_{sd} presents a moderate level of satisfaction in all the rounds because its only criterion is to minimize group disagreement. As a result, It could not be able to maximize the satisfaction level of the users. Finally, an improved trend is visible after combining satisfaction with m-proportionality, and R_{smprop} generates a better performance by creating better satisfaction scores than R_{mprop}. However, R_{smprop} can not exceed the satisfaction level of R_s, but it indicates that the recommended items are not biased by eliminating the risk of neglecting certain users of the group and ensuring fairness. While concentrating on the disagreement scores it is important to note that both R_s and R_{mprop} result in a higher group disagreement score compared to R_{sd} and R_{smprop} (Figure 1(b)). Additionally, R_{smprop} generates slightly lower disagreement scores compared to R_{sd}. This outcome aligns with expectations since R_{smprop} considers both group satisfaction and m-proportionality, resulting in a more balanced set of recommended items.

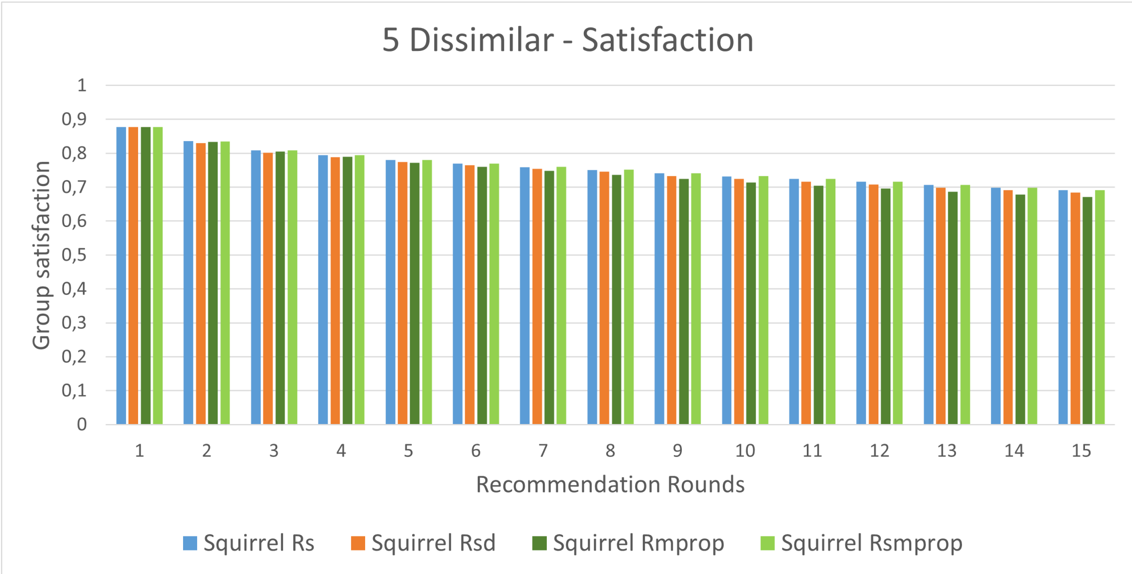
The results for the 5-Diss group settings are shown in Figures 5.2(a) and 5.2(b). From the figures, it can be seen that all the reward functions perform similarly for group satisfaction. Similar to the 4+1 group settings, R_s consistently outperforms other functions in all the recommendation rounds by prioritizing group satisfaction. However, once again while preserving each user's choice in the group recommen-



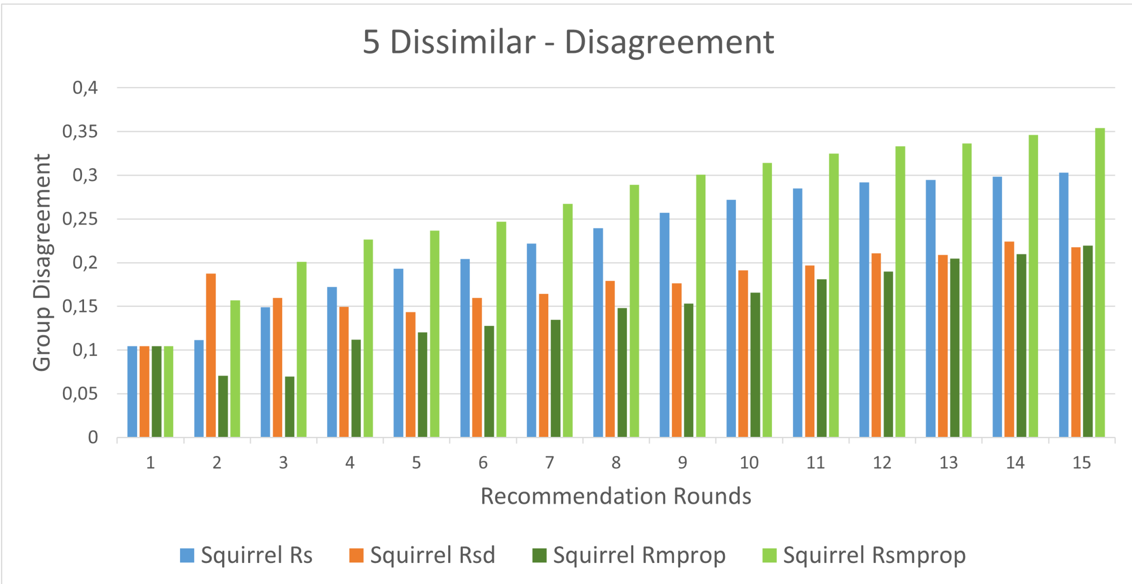
(a)

(b)

Figure 5.1 *Group Satisfaction and Disagreement for SQUIRREL –* R_s, R_{sd}, R_{mprop}, R_{smprop} *in 4+1 groups.*



(a)



(b)

Figure *5.2 Group Satisfaction and Disagreement for SQUIRREL* – R_s, R_{sd}, R_{mprop}, R_{smprop} *in 5-Diss groups.*

dation list, R_{smprop} retains a higher satisfaction score. On the other hand, this group settings result in a somewhat lower satisfaction score for R_{mprop}. As all the users of the group are dissimilar to each other, thus it is hard to maintain a good satisfaction score for every user. In disagreement scores, R_{sd} and R_{mprop} perform better in reducing disagreement, with R_{sd} having slightly higher values in most of the iterations. Naturally, R_{mprop}'s main goal is to always recommend the best items to each consumer, ultimately minimizing any disagreement among users. In the 5-Diss group settings, with all users having distinct preferences, leads to R_{smprop} yielding the highest group disagreement score. This condition complicates the task of reducing disagreement while simultaneously enhancing satisfaction by suggesting a minimum of two preferred items for each user.

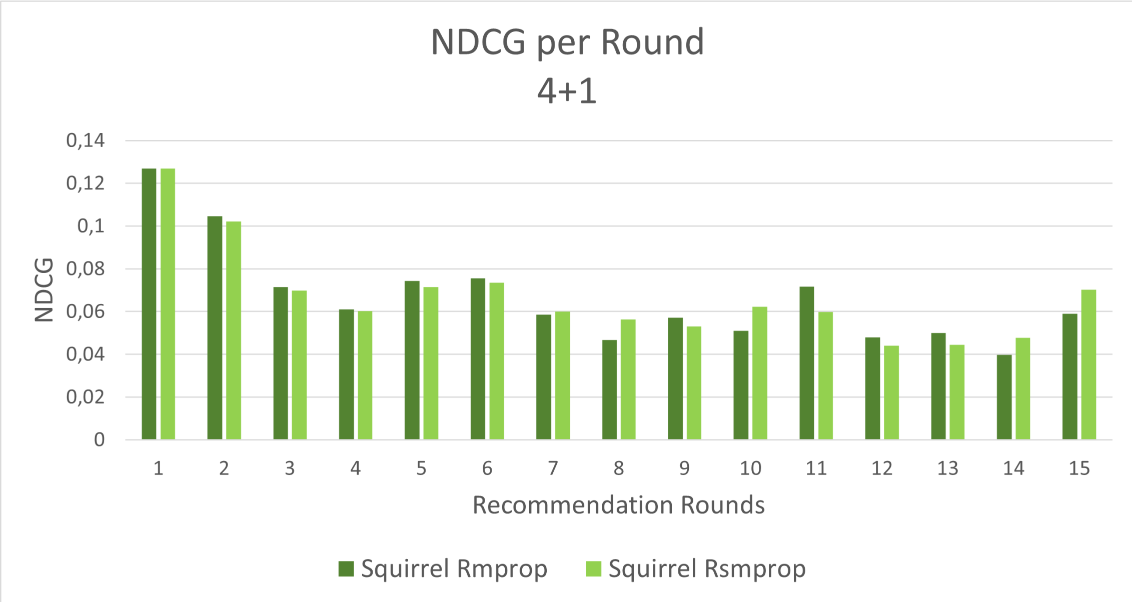
5.2.4 Quality of Recommendations

In addition to the group satisfaction and disagreement metrics, we utilize the Normalized Discounted Cumulative Gain (NDCG) to evaluate how well our reward systems work. This NDCG value measures how well the recommended items match the group's preferences by comparing them to each user's individual preferences. Table 5.1 presents the NDCG scores for all SQUIRREL rewards, including those incorporating the new fairness reward functions. These scores represent the average performance over 15 rounds of recommendations.

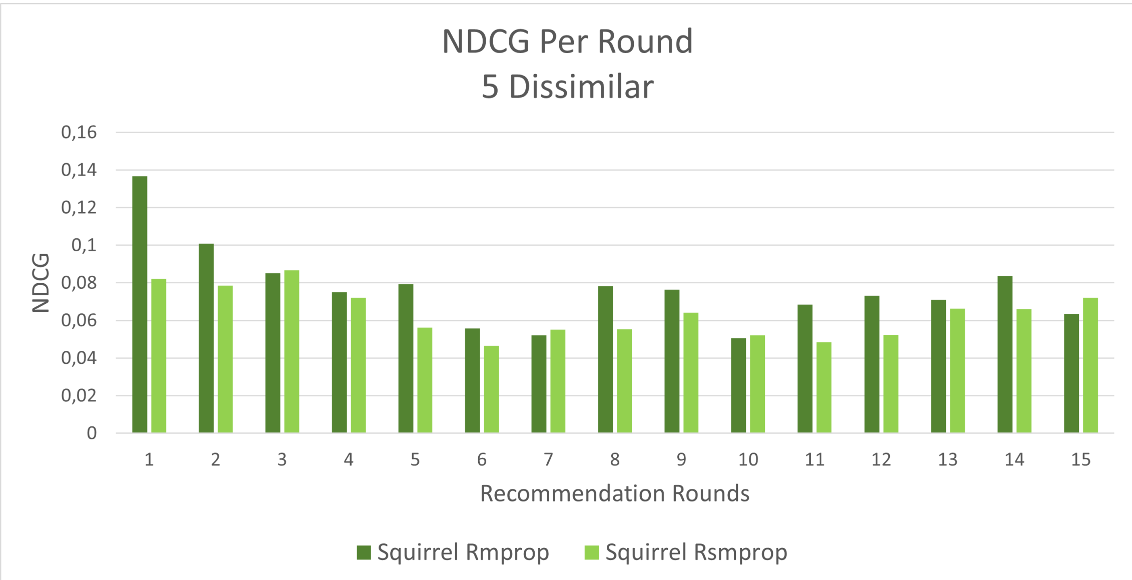
Table 5.1 *NDCG values for the SQUIRREL models in all test scenarios.*

	R_s	R_{sd}	R_{mprop}	R_{smprop}
4+1	0.052	0.054	0.066	0.067
5-Diss	0.060	0.054	0.076	0.064

This analysis reveals that the newly introduced fairness-based SQUIRREL models, namely R_{mprop} and R_{smprop}, outperform the others, demonstrating their capacity to find more pertinent items for the groups. Figure 5.3 illustrates the NDCG scores for these new rewards, showing a higher performance in the initial recommendation rounds with some variation in the subsequent rounds. This variation can be attributed to the changing dataset sparsity as new chunks are added and the exclusion of top items from consideration after each round, contributing to the decline in performance.



(a)



(b)

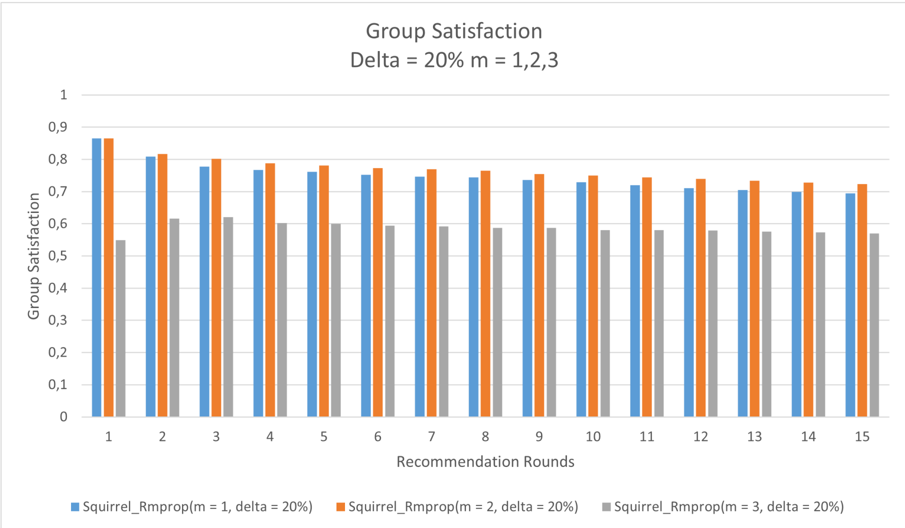
Figure 5.3 *NDCG Values per recommendation round for* $SQUIRREL - R_{mprop}$, R_{smprop} *in the 4+1 and 5-Diss test scenario.*

5.3 Explanation of Fairness Rewards attributes

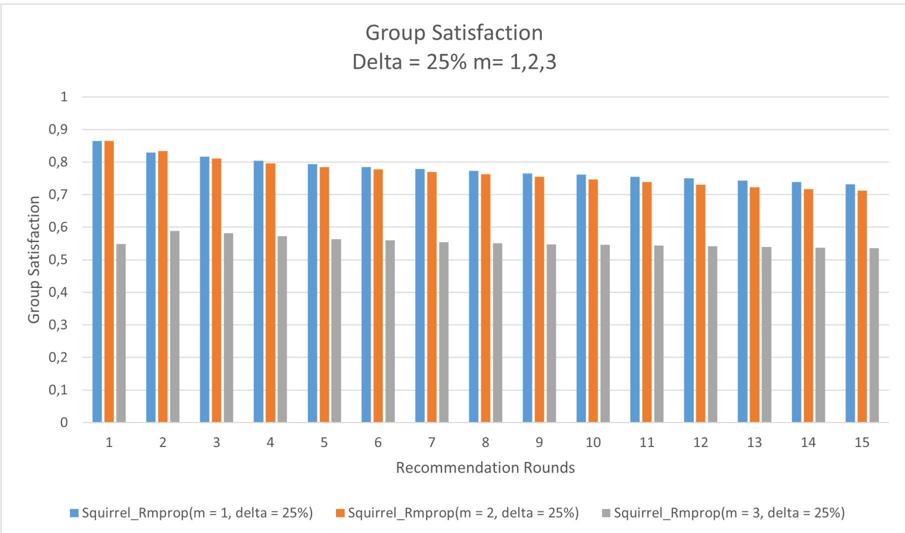
During the experiments with the new fairness rewards R_{mprop} and R_{smprop}, it was crucial to decide the best value for m and Δ. Depending on the group configuration and the number of items in the group recommendation lists for each round, these attributes may have varying values. In this work, we select 10 items for the final recommendation list and 5 members for each group. To determine the ideal values of Δ and m we calculated the value of group satisfaction and disagreement for different values of m and Δ for both the group settings, (4+1) and (5Diss). Here, the Δ value varies from 15% to 25% and m takes values of 1,2 and 3. Figures 5.4 and 5.5 show the group Satisfaction and Disagreement of SQUIRREL $- R_{mprop}$ in 4+1 group settings for different values of m and Δ. Here we can see that, when the number of m rises, group satisfaction falls for all values of Δ. When *m=1 and 2*, we observe higher values in group satisfaction (Figure 5.4 (a,b,c)). However, for $m = 1$, it shows a slightly better satisfaction score compared to $m = 2$. In contrast, when m = 3, a sudden drop in satisfaction is observed. The main reason can be found in the definition of the attributes and the current group settings of the SQUIRREL model. In our fairness rewards, m represents the minimum quantity of items from the top Δ% of the users' individual lists, that must need to be on the list of recommendations for the group. The model deals with a 5-member group and generates 10 items for them, then it is easy to accommodate at least 1 to 2 items for every group member through all the values of Δ. However, it became challenging to recommend 3 items for each user while generating 10 items for a 5 members group. A similar pattern is observed for the group disagreement as well (Figure 5.5 (a,b,c)). We can see that, the disagreement values are less *m=1 and 2*. Also, it is very high when $m = 3$.

The effect of Δ is also observed in the group satisfaction and disagreement. When $\Delta = 15\%$ it was easy to recommend one item for all. However, we can see a slight improvement in group satisfaction value when we increase the value of Δ to 20% and 25%. In contrast, we can see relatively small disagreement scores when $\Delta = 15\%$ and 20%. However, $\Delta = 25\%$ shows slightly more disagreement scores compared to other values of Δ. This is because, in the SQUIRREL model, we don't recommend the items that are already recommended in any previous rounds. As a result, it became difficult to keep a lower disagreement score, although it is expected as Δ is increased. After carefully observing all the scenarios mentioned above, it was decided that $m = 2$ and $\Delta = 20\%$ are the most suitable values in our current experimental scenario.

(a)

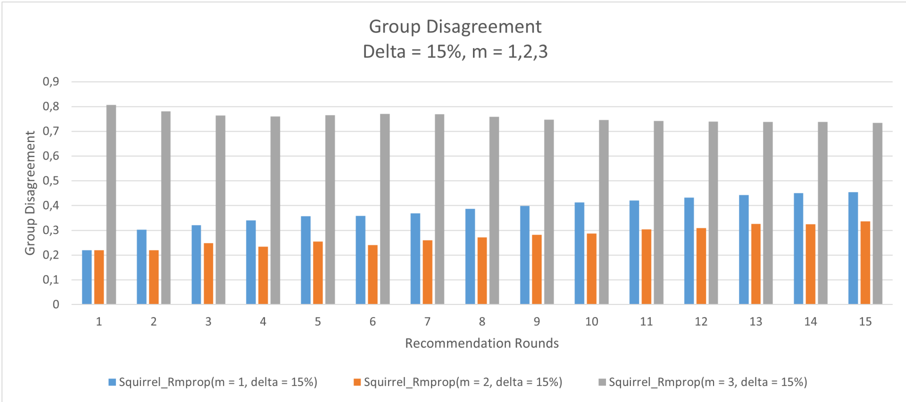


(b)

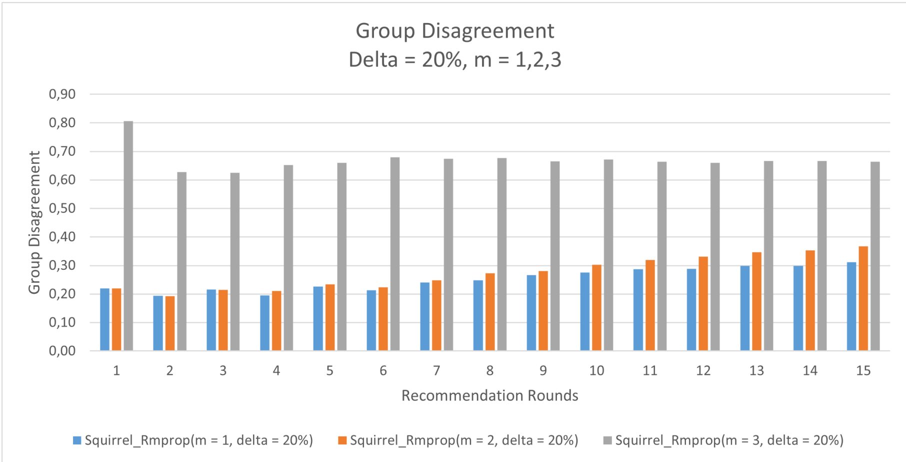


(c)

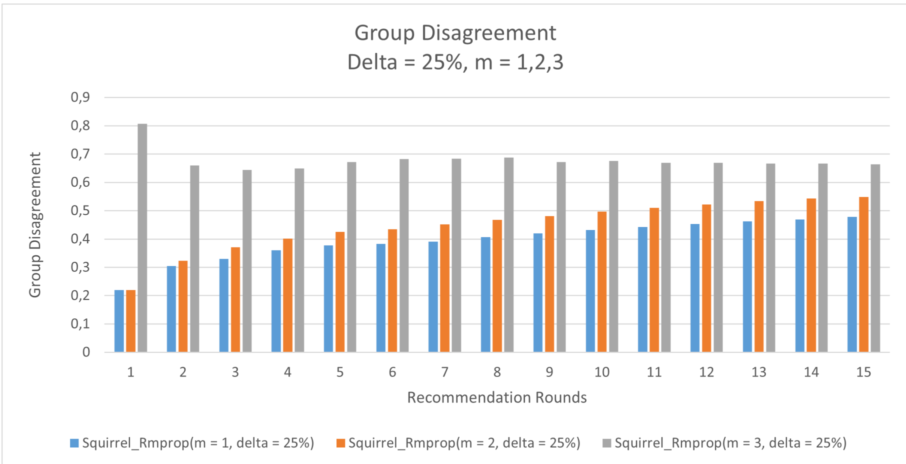
***Figure* 5.4** *Group Satisfaction of SQUIRREL $-$ R_{mprop} in 4+1 group settings for different values of m and Δ.*



(a)



(b)



(c)

Figure 5.5 *Group Disagreement of SQUIRREL* $- R_{mprop}$ *in 4+1 group settings for different values of* m *and* Δ.

6 Conclusions

This study has focused on fairness, one of the major research problems in the field of recommendation systems for groups. It addresses the need for fairness criteria in sequential recommendation systems and proposes new fairness measures. To effectively evaluate the new fairness measures it expands on a reinforcement learning framework for sequential group recommendations called SQUIRREL. The SQUIRREL approach focuses only on compiling group members' recommendation lists into a single list, treating the single-user recommendation system as precalculated knowledge. This aggregation is achieved through various aggregation methods. Each group member's satisfaction level determines the model's state, and there are two different reward functions. It chooses the best group recommendation strategy dynamically depending on the group's condition at the time. However, the configuration of the SQUIRREL model is flexible, allowing for the application of additional ranking methods, the definition of different states, and the consideration of various reward functions. In this work, the state definitions and actions of the original SQUIRREL model remain intact. Two new fairness-based reward functions called M-proportionality (R_{mprop}) and the combination of Satisfaction and M-proportionality (R_{smprop}) are used to implement fairness-based recommendations.

This work creates a new dataset that precisely matches the issue specification, derived from the 20M MovieLens dataset. Furthermore, every possible aggregating technique is used as an action. As a reward function, two new fairness reward functions along with the old ones are used. The model is trained using various group formations, representing diverse real world scenarios for group recommendations. During the training period, the model learns which aggregation method maximizes the rewards for the current condition of the group. After that, different sets of data are used to test the model. The group Satisfaction and group disagreement metrics are used to evaluate its performance. Here, we discovered that there is a nice balance between group disagreement and satisfaction with the new fairness-based measurements. In addition, the recommendation's quality is evaluated by calculating the DFH and NDCG values for the newly configured SQUIRREL model. From the results, it can be seen that SQUIRREL delivers high-quality recommendations for fairness-based reward functions, as evidenced by its excellent NDCG and DFH scores.

This study opens doors for several exciting future research directions that can further enhance fairness in sequential group recommendations. The current work focuses only on M-proportionality fairness measures. Future research could explore incorporating fairness for other user attributes like age, location, or genre preferences. This would require defining new fairness metrics and potentially modifying the reward functions to consider these additional factors. This work establishes that fairness and group satisfaction can be balanced. In the future, we could delve into explaining the recommendations to users. This explanation should not only highlight the chosen items but also transparently showcase how fairness considerations influenced the selection.

7 References

[1] Maria Stratigi, Evaggelia Pitoura, and Kostas Stefanidis. "SQUIRREL: A framework for sequential group recommendations through reinforcement learning". In: *Information Systems* 112 (2023), p. 102128. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2022.102128. URL: https://www.sciencedirect.com/science/article/pii/S0306437922001065.

[2] Judith Masthoff. "Group Recommender Systems: Combining Individual Models". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, et al. Boston, MA: Springer US, 2011, pp. 677–702. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_21. URL: https://doi.org/10.1007/978-0-387-85820-3_21.

[3] Mike Gartrell, Xinyu Xing, Qin Lv, et al. "Enhancing group recommendation by incorporating social relationship interactions". In: *Proceedings of the 2010 ACM International Conference on Supporting Group Work*. GROUP '10. Sanibel Island, Florida, USA: Association for Computing Machinery, 2010, pp. 97–106. ISBN: 9781450303873. DOI: 10.1145/1880071.1880087. URL: https://doi.org/10.1145/1880071.1880087.

[4] Judith Masthoff. "Group Recommender Systems: Aggregation, Satisfaction and Group Attributes". In: Jan. 2015, pp. 743–776. ISBN: 978-1-4899-7636-9. DOI: 10.1007/978-1-4899-7637-6_22.

[5] Anthony Jameson and Barry Smyth. "Recommendation to Groups". In: *The Adaptive Web: Methods and Strategies of Web Personalization*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 596–627. ISBN: 978-3-540-72079-9. DOI: 10.1007/978-3-540-72079-9_20. URL: https://doi.org/10.1007/978-3-540-72079-9_20.

[6] Joseph Mccarthy and Theodore Anagnost. "MUSICFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts". In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (Jan. 2000). DOI: 10.1145/289444.289511.

[7] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, et al. "TV Program Recommendation for Multiple Viewers Based on user Profile Merging". In: *User Model. User-Adapt. Interact.* 16 (Mar. 2006), pp. 63–82. DOI: 10.1007/s11257-006-9005-6.

[8] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, et al. "Group recommendation: semantics and efficiency". In: *Proc. VLDB Endow.* 2.1 (Aug. 2009), pp. 754–765. ISSN: 2150-8097. DOI: 10.14778/1687627.1687713. URL: https://doi.org/10.14778/1687627.1687713.

[9] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. "Group recommendations with rank aggregation and collaborative filtering". In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10. Barcelona, Spain: Association for Computing Machinery, 2010, pp. 119–126. ISBN: 9781605589060. DOI: 10.1145/1864708.1864733. URL: https://doi.org/10.1145/1864708.1864733.

[10] Eirini Ntoutsi, Kostas Stefanidis, Kjetil Norvag, et al. "Fast Group Recommendations by Applying User Clustering". In: Oct. 2012. ISBN: 978-3-642-34001-7. DOI: 10.1007/978-3-642-34002-4_10.

[11] Mark O'Connor, Dan Cosley, Joseph Konstan, et al. "PolyLens: A Recommender System for Groups of Users". In: May 2007, pp. 199–218. ISBN: 0-7923-7162-3. DOI: 10.1007/0-306-48019-0_11.

[12] Quan Yuan, Gao Cong, and Chin-Yew Lin. "COM: A generative model for group recommendation". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2014). DOI: 10.1145/2623330.2623616.

[13] Xiangnan He, Zhankui He, Xiaoyu Du, et al. "Adversarial Personalized Ranking for Recommendation". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '18. Ann Arbor, MI, USA: Association for Computing Machinery, 2018, pp. 355–364. ISBN: 9781450356572. DOI: 10.1145/3209978.3209981. URL: https://doi.org/10.1145/3209978.3209981.

[14] Hongzhi Yin, Qinyong Wang, Kai Zheng, et al. "Social Influence-Based Group Representation Learning for Group Recommendation". In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 2019, pp. 566–577. DOI: 10.1109/ICDE.2019.00057.

[15] Amirali Salehi-Abari and Craig Boutilier. "Preference-oriented Social Networks: Group Recommendation and Inference". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys '15. Vienna, Austria: Association for Computing Machinery, 2015, pp. 35–42. ISBN: 9781450336925. DOI: 10.1145/2792838.2800190. URL: https://doi.org/10.1145/2792838.2800190.

[16] Lucas Vinh Tran, Tuan-Anh Pham, Yi Tay, et al. "Interact and Decide: Medley of Sub-Attention Networks for Effective Group Recommendation". In: July 2019, pp. 255–264. ISBN: 978-1-4503-6172-9. DOI: 10.1145/3331184.3331251.

[17] Dong Qin, Xiangmin Zhou, Lei Chen, et al. "Dynamic Connection-Based Social Group Recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* PP (Nov. 2018), pp. 1–1. DOI: 10.1109/TKDE.2018.2879658.

[18] "A group recommendation system for online communities". English. In: *International Journal of Information Management* 30.3 (June 2010), pp. 212–219. ISSN: 0268-4012. DOI: 10.1016/j.ijinfomgt.2009.09.006.

[19] "Fairness-aware group recommendation with pareto-efficiency". English (US). In: *RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems*. RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems. Association for Computing Machinery, Inc, Aug. 2017, pp. 107–115. DOI: 10.1145/3109859.3109887.

[20] Dimitris Sacharidis. "Top-N group recommendations with fairness". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC '19. Limassol, Cyprus: Association for Computing Machinery, 2019, pp. 1663–1670. ISBN: 9781450359337. DOI: 10.1145/3297280.3297442. URL: https://doi.org/10.1145/3297280.3297442.

[21] Mesut Kaya, Derek Bridge, and Nava Tintarev. "Ensuring Fairness in Group Recommendations by Rank-Sensitive Balancing of Relevance KEYWORDS group recommendations, fairness". In: Sept. 2020. DOI: 10.1145/3383313.3412232.

[22] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. *Sequence-Aware Recommender Systems*. 2018. arXiv: 1802.08452 [cs.IR].

[23] Chen Cheng, Haiqin Yang, Michael R. Lyu, et al. "Where you like to go next: successive point-of-interest recommendation". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China: AAAI Press, 2013, pp. 2605–2611. ISBN: 9781577356332.

[24] Defu Lian, Vincent Zheng, and Xing Xie. "Collaborative filtering meets next check-in location prediction". In: May 2013, pp. 231–232. DOI: 10.1145/2487788.2487907.

[25] Qiang Liu, Shu Wu, Liang Wang, et al. "Predicting the next location: a recurrent model with spatial and temporal contexts". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, 2016, pp. 194–200.

[26] Ilija Ilievski and Sujoy Roy. "Personalized news recommendation based on implicit feedback". In: *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*. NRS '13. Kowloon, Hong Kong: Association for Computing Machinery, 2013, pp. 10–15. ISBN: 9781450323024. DOI: 10.1145/2516641.2516644. URL: https://doi.org/10.1145/2516641.2516644.

[27] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, et al. "Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys '16. Boston, Massachusetts, USA: Association for Computing Machinery, 2016, pp. 241–248. ISBN: 9781450340359. DOI: 10.1145/2959100.2959167. URL: https://doi.org/10.1145/2959100.2959167.

[28] Negar Hariri, Bamshad Mobasher, and Robin Burke. "Context-aware music recommendation based on latenttopic sequential patterns". In: *Proceedings of the Sixth ACM Conference on Recommender Systems*. RecSys '12. Dublin, Ireland: Association for Computing Machinery, 2012, pp. 131–138. ISBN: 9781450312707. DOI: 10.1145/2365952.2365979. URL: https://doi.org/10.1145/2365952.2365979.

[29] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. "Adaptation and Evaluation of Recommendations for Short-term Shopping Goals". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys '15. Vienna, Austria: Association for Computing Machinery, 2015, pp. 211–218. ISBN: 9781450336925. DOI: 10.1145/2792838.2800176. URL: https://doi.org/10.1145/2792838.2800176.

[30] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, et al. "Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks". In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. RecSys '17. Como, Italy: Association for Computing Machinery, 2017, pp. 130–137. ISBN: 9781450346528. DOI: 10.1145/3109859.3109896. URL: https://doi.org/10.1145/3109859.3109896.

[31] Casper Hansen, Christian Hansen, Lucas Maystre, et al. "Contextual and Sequential User Embeddings for Large-Scale Music Recommendation". In: Sept. 2020, pp. 53–62. DOI: 10.1145/3383313.3412248.

[32] Rodrigo Borges and Kostas Stefanidis. "Enhancing Long Term Fairness in Recommendations with Variational Autoencoders". In: Nov. 2019, pp. 95–102. DOI: 10.1145/3297662.3365798.

[33] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. “Fairness in rankings and recommendations: an overview”. In: *The VLDB Journal* 31.3 (Oct. 2021), pp. 431–458. ISSN: 1066-8888. DOI: 10.1007/s00778-021-00697-y. URL: https://doi.org/10.1007/s00778-021-00697-y.

[34] Rodrigo Borges and Kostas Stefanidis. “On mitigating popularity bias in recommendations via variational autoencoders”. In: Mar. 2021, pp. 1383–1389. DOI: 10.1145/3412841.3442123.

[35] Rodrigo Borges and Kostas Stefanidis. “F2VAE: a framework for mitigating user unfairness in recommendation systems”. In: *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*. SAC ’22. Virtual Event: Association for Computing Machinery, 2022, pp. 1391–1398. ISBN: 9781450387132. DOI: 10.1145/3477314.3507152. URL: https://doi.org/10.1145/3477314.3507152.

[36] Rodrigo Borges and Kostas Stefanidis. “On Measuring Popularity Bias in Collaborative Filtering Data”. English. In: *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference*. Vol. 2578. CEUR Workshop Proceedings. jufoid=53269; EDBT/ICDT Workshops ; Conference date: 01-01-2020. CEUR-WS.org, 2020.

[37] Jiarui Qin, Kan Ren, Yuchen Fang, et al. “Sequential Recommendation with Dual Side Neighbor-based Collaborative Relation Modeling”. In: *Proceedings of the 13th International Conference on Web Search and Data Mining*. WSDM ’20. Houston, TX, USA: Association for Computing Machinery, 2020, pp. 465–473. ISBN: 9781450368223. DOI: 10.1145/3336191.3371842. URL: https://doi.org/10.1145/3336191.3371842.

[38] Wen Wang, Wei Zhang, Jun Rao, et al. “Group-Aware Long- and Short-Term Graph Representation Learning for Sequential Group Recommendation”. In: July 2020, pp. 1449–1458. DOI: 10.1145/3397271.3401136.

[39] Maria Stratigi, Jyrki Nummenmaa, Evaggelia Pitoura, et al. “Fair sequential group recommendations”. In: Mar. 2020, pp. 1443–1452. DOI: 10.1145/3341105.3375766.

[40] Maria Stratigi, Evaggelia Pitoura, Jyrki Nummenmaa, et al. “Sequential group recommendations based on satisfaction and disagreement scores”. In: *J. Intell. Inf. Syst.* 58.2 (Apr. 2022), pp. 227–254. ISSN: 0925-9902. DOI: 10.1007/s10844-021-00652-x. URL: https://doi.org/10.1007/s10844-021-00652-x.

[41] Judith Masthoff. “Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers”. In: *User Model. User-Adapt. Interact.* 14 (Feb. 2004), pp. 37–85. DOI: 10.1023/B:USER.0000010138.79319.fd.

[42] A. Piliponyte, Francesco Ricci, and J. Koschwitz. “Sequential music recommendations for groups by balancing user satisfaction”. In: 997 (Jan. 2013).

[43] Yunqi Li, Hanxiong Chen, Shuyuan Xu, et al. “Fairness in Recommendation: Foundations, Methods, and Applications”. In: *ACM Trans. Intell. Syst. Technol.* 14.5 (Oct. 2023). ISSN: 2157-6904. DOI: 10.1145/3610302. URL: https://doi.org/10.1145/3610302.

[44] Caitlin Kuhlman and Elke Rundensteiner. “Rank aggregation algorithms for fair consensus”. In: *Proc. VLDB Endow.* 13.12 (July 2020), pp. 2706–2719. ISSN: 2150-8097. DOI: 10.14778/3407790.3407855. URL: https://doi.org/10.14778/3407790.3407855.

[45] Lin Xiao, Zhang Min, Zhang Yongfeng, et al. “Fairness-Aware Group Recommendation with Pareto-Efficiency”. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. RecSys ’17. event-place: Como, Italy. New York, NY, USA: ACM, 2017, pp. 107–115. ISBN: 978-1-4503-4652-8. DOI: 10.1145/3109859.3109887. URL: http://doi.acm.org/10.1145/3109859.3109887 (visited on 05/17/2019).

[46] Lucas Augusto Montalvão Costa Carvalho and Hendrik Teixeira Macedo. “Users’ satisfaction in recommendation systems for groups: an approach based on noncooperative games”. In: *Proceedings of the 22nd International Conference on World Wide Web*. WWW ’13 Companion. Rio de Janeiro, Brazil: Association for Computing Machinery, 2013, pp. 951–958. ISBN: 9781450320382. DOI: 10.1145/2487788.2488090. URL: https://doi.org/10.1145/2487788.2488090.

[47] J. Neumann and O. Morgenstern. “Theory of games and economic behavior (60th anniversary commemorative edition)”. In: *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)* 51 (Jan. 2007), pp. 1–741.

[48] Lihi Naamani Dery, Meir Kalech, Lior Rokach, et al. “Iterative voting under uncertainty for group recommender systems”. In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys ’10. Barcelona, Spain: Association for Computing Machinery, 2010, pp. 265–268. ISBN: 9781605589060. DOI: 10.1145/1864708.1864763. URL: https://doi.org/10.1145/1864708.1864763.

[49] Francesca Guzzi, Francesco Ricci, and Robin Burke. “Interactive multi-party critiquing for group recommendation”. In: *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys ’11. Chicago, Illinois, USA: Association for Computing Machinery, 2011, pp. 265–268. ISBN: 9781450306836. DOI: 10.1145/2043932.2043980. URL: https://doi.org/10.1145/2043932.2043980.

[50] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, et al. “Fairness in Package-to-Group Recommendations”. In: *Proceedings of the 26th International Conference on World Wide Web* (2017). URL: https://api.semanticscholar.org/CorpusID:1171836.

[51] M. Mehdi Afsar, Trafford Crump, and Behrouz Far. “Reinforcement Learning based Recommender Systems: A Survey”. In: *ACM Comput. Surv.* 55.7 (Dec. 2022). ISSN: 0360-0300. DOI: 10.1145/3543846. URL: https://doi.org/10.1145/3543846.

[52] Nima Taghipour, Ahmad Kardan, and Saeed Ghidary. “Usage-based web recommendations: a reinforcement learning approach”. In: Oct. 2007, pp. 113–120. DOI: 10.1145/1297231.1297250.

[53] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, et al. “DRN: A Deep Reinforcement Learning Framework for News Recommendation”. In: *Proceedings of the 2018 World Wide Web Conference*. WWW ’18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 167–176. ISBN: 9781450356398. DOI: 10.1145/3178876.3185994. URL: https://doi.org/10.1145/3178876.3185994.

[54] Liwei Huang, Mingsheng Fu, Fan Li, et al. “A deep reinforcement learning based long-term recommender system”. In: *Know.-Based Syst.* 213.C (Feb. 2021). ISSN: 0950-7051. DOI: 10.1016/j.knosys.2020.106706. URL: https://doi.org/10.1016/j.knosys.2020.106706.

[55] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, et al. “Deep Reinforcement Learning for List-wise Recommendations”. In: *CoRR* abs/1801.00209 (2018). arXiv: 1801.00209. URL: http://arxiv.org/abs/1801.00209.

[56] Zhang Yuyan, Su Xiayao, and Liu Yong. “A Novel Movie Recommendation System Based on Deep Reinforcement Learning with Prioritized Experience Replay”. In: Oct. 2019, pp. 1496–1500. DOI: 10.1109/ICCT46805.2019.8947012.

[57] O Moling, Linas Baltrunas, and Francesco Ricci. “Optimal radio channel recommendations with explicit and implicit feedback”. eng. In: New York, NY: ACM, 2012, p. 8. ISBN: 978-1-4503-1270-7. DOI: 10.1145/2365952.2365971.

[58] Guy Shani, Ronen Brafman, and David Heckerman. “An MDP-based recommender system”. In: *Journal of Machine Learning Research* 6 (Dec. 2012).

[59] Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, et al. “Recommending packages with validity constraints to groups of users”. In: *Knowl. Inf. Syst.* 54.2 (Feb. 2018), pp. 345–374. ISSN: 0219-1377. DOI: 10.1007/s10115-017-1082-9. URL: https://doi.org/10.1007/s10115-017-1082-9.

[60] F. Maxwell Harper and Joseph A. Konstan. “The MovieLens Datasets: History and Context”. In: *ACM Trans. Interact. Intell. Syst.* 5.4 (Dec. 2015). ISSN: 2160-6455. DOI: 10.1145/2827872. URL: https://doi.org/10.1145/2827872.

[61] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, et al. “GroupLens: an open architecture for collaborative filtering of netnews”. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. CSCW ’94. Chapel Hill, North Carolina, USA: Association for Computing Machinery, 1994, pp. 175–186. ISBN: 0897916891. DOI: 10.1145/192844.192905. URL: https://doi.org/10.1145/192844.192905.

[62] Christian Desrosiers and George Karypis. “A Comprehensive Survey of Neighborhood-Based Recommendation Methods”. In: Jan. 2011, pp. 107–144. ISBN: 978-0-387-85819-7. DOI: 10.1007/978-0-387-85820-3_4.

[63] Xiaoyuan Su and Taghi M. Khoshgoftaar. “A survey of collaborative filtering techniques”. In: *Adv. in Artif. Intell.* 2009 (Jan. 2009). ISSN: 1687-7470. DOI: 10.1155/2009/421425. URL: https://doi.org/10.1155/2009/421425.

[64] Yifan Wang, Weizhi Ma, Min Zhang, et al. “A Survey on the Fairness of Recommender Systems”. In: *ACM Trans. Inf. Syst.* 41.3 (Feb. 2023). ISSN: 1046-8188. DOI: 10.1145/3547333. URL: https://doi.org/10.1145/3547333.