Technical University of Crete

Department of Electrical and Computer Engineering

Item Fairness in Network Friendly Recommendation Systems

Author:
Panagiotis Angelos Ganotis

Thesis Committee:
Prof. Spyropoulos Thrasyvoulos
Prof. Liavas Athanasios
Prof. Chialkiadakis Georgios

A thesis submitted in fulfillment of the requirements for the diploma of

Electrical and Computer Engineer

March 16, 2024

Acknowledgements

I would like to express my sincere gratitude to everyone who has contributed to the completion of this thesis. First and foremost, I am deeply thankful to my supervisor, Thrasyvoulos Spyropoulos, for his invaluable guidance and support throughout this journey. His expertise, insights, and mentorship have been instrumental in shaping the direction of this thesis. Furthermore, I would like to thank Prof. Georgios Chalkiadakis and Prof. Athanasios Liavas for being members of the thesis committee.

My heartfelt appreciation goes to my colleagues, friends, and family for their unwavering support, encouragement, and understanding throughout this challenging yet rewarding endeavor. Their encouragement, moral support, and "occasional distractions" have kept me motivated and inspired during the ups and downs of this academic journey.

Contents

Abstract

Recommendation Systems (RS) play a crucial role in shaping narratives and influencing people's choices. This technological field has seen a lot of research in recent years, both due to its utility to users and also due to its ethical implications. Most content and social media platforms use recommendation systems to give users the most relevant and engaging options based on their past preferences. Recent work suggests integrating network performance into the design of recommendation system algorithms, demonstrating considerable potential. However, network-friendly adaptations of RS algorithms may create unfairness for users and content providers. A Network-friendly Recommendation System also introduces bias toward (a smaller pool of) low-cost content, raising fairness concerns for both consumers and creators. Fairness is now considered a complementary optimization dimension to achieve a 'win-win-win' situation for network/content providers, users, and content creators.

This thesis focuses on exploring the creation of 'content bubbles' as a specific form of unfairness, which has not been extensively studied in the problem before. At a high level, a 'content bubble' implies that the selection of items suggested to a user (or group of users) is less diversified in order to facilitate network cost reduction. The main contribution of this thesis is to define appropriate metrics to measure 'content bubble' and answer the following questions: i) Do NF-RS algorithms foster content bubbles, and ii) Do current fairness metrics effectively address this issue? Our results have indicated that the existing metrics do not fully capture the 'content bubble' problem and moreover, the tolerance of unfairness in order to establish high network gains may cause recommendations to lack diversity, strengthening the 'content bubble' effect.

Περίληψη

Τα Συστήματα Συστάσεων (ΣΣ) διαδραματίζουν κρίσιμο ρόλο στη διαμόρφωση των επιλογών των ανθρώπων. Αυτό το τεχνολογικό πεδίο είναι υπό αρκετή έρευνα τα τελευταία χρόνια, τόσο λόγω της χρησιμότητάς του για τους χρήστες όσο και λόγω των ηθικών του επιπτώσεων. Οι περισσότερες πλατφόρμες περιεχομένου και κοινωνικών μέσων χρησιμοποιούν συστήματα συστάσεων για να παρέχουν στους χρήστες τις πιο σχετικές και ελκυστικές επιλογές με βάση τις προηγούμενες προτιμήσεις τους. Οι τελευταίες ερευνητικές εργασίες προτείνουν ότι η ενσωμάτωση της απόδοσης του δικτύου στο σχεδιασμό αλγορίθμων συστημάτων συστάσεων, φαίνεται να έχει σημαντικές προοπτικές. Ωστόσο, οι **Φ**ιλικές προς το **Δ**ίκτυο προσαρμογές των αλγορίθμων **Σ**υστημάτων **Σ**υστάσεων (**ΦΔ-ΣΣ**) ενδέχεται να δημιουργήσουν αδικία για χρήστες και δημιουργούς περιεχομένου. Ένα σύστημα συστάσεων φιλικό προς το δίκτυο εισάγει επίσης μεροληψία προς (ένα μικρότερο σύνολο) περιεχομένου χαμηλού κόστους, αυξάνοντας τις ανησυχίες περί δικαιοσύνης τόσο για τους καταναλωτές όσο και για τους δημιουργούς. Η δικαιοσύνη θεωρείται πλέον μία συμπληρωματική διάσταση βελτιστοποίησης για την επίτευξη μιας κατάστασης ''τριπλού κέρδους'' για παρόχους δικτύου/περιεχομένου, χρήστες και δημιουργούς περιεχομένου.

Αυτή η διατριβή εστιάζει στη διερεύνηση της δημιουργίας «φυσαλίδων περιεχομένου» ως συγκεκριμένη μορφή αδικίας, η οποία δεν έχει μελετηθεί εκτενώς ακόμη. Με απλά λόγια, μια «φούσκα περιεχομένου» σημαίνει ότι η επιλογή των προτεινόμενων αντικειμένων προς έναν χρήστη (ή ομάδα χρηστών) είναι λίγο διαφοροποιημένη προκειμένου να διευκολυνθεί η μείωση στο κόστος του δικτύου. Η κύρια συμβολή αυτής της διπλωματικής εργασίας είναι ο καθορισμός κατάλληλων μετρικών ώστε να μετρηθεί η «φούσκα περιεχομένου» και να απαντηθούν οι ακόλουθες ερωτήσεις: 1)Αν οι αλγόριθμοι ΦΔ-ΣΣ ενισχύουν τις φυσαλίδες περιεχομένου και 2) οι τρέχουσες μετρικές δικαιοσύνης καταγράφουν αποτελεσματικά αυτό το ζήτημα· Τα αποτελέσματα μας έδειξαν ότι οι υπάρχουσες μετρικές δεν αποτυπώνουν πλήρως το πρόβλημα της «φούσκας περιεχομένου» και επιπλέον, η ανοχή στην αδικία προκειμένου να δημιουργηθούν υψηλά κέρδη δικτύου μπορεί να προκαλέσει συστάσεις με έλλειψη ποικιλίας, ενισχύοντας το φαινόμενο της «φούσκας περιεχομένου».

Chapter 1

Introduction

In this chapter, we will provide a quick glimpse of what this thesis is all about. Starting from what a Recommendation system is and then finally presenting the objective of this thesis.

1.1 The art of Recommendations Systems

We will first present the functionality of a recommendation system and where it is used in recent days.

Recommendation systems are software algorithms designed to provide personalized suggestions or recommendations to consumers based on their distinct interests. These suggestions may refer to products, services, content, and other resources that are provided on a certain platform or system. Recommendation systems are utilized on various online platforms, such as e-commerce websites, streaming services, social networking platforms, and content websites, with the aim of enhancing customer's experience with the service.

The recommendations are generated by a filtering procedure that produces results based on the user's interests and past selections. Two fundamental algorithms in recommendation systems are *Collaborative* and *Content-Based* filtering. The former focuses more on the user's preferences, and the latter relies more on information about the content and its features. Given the complex structure of present-day data, it is unusual to employ these strategies individually. Instead, these concepts are combined in a methodology known as *Hybrid Filtering*. Netflix, YouTube, and other renowned platforms employ these types of algorithms to deliver the optimal user experience and maintain user engagement.

All the above platforms are using the network in order to provide their services and handle all the data that will be processed by their recommendation algorithms. Although there has been a lot of research and sophistication contained in many RS algorithms, these often do not consider what the network cost is of fetching the recommended content to the user. This problem has caught the attention of the scientific community, and recent work has emerged proposing *network-friendly* recommendation algorithms that will aim to maintain a good recommendation quality while also trying to reduce the network cost.

1.2 Network Friendly Recommendation Systems

The concept of including network performance in RSs has been presented to the scientific community in recent years. Network-friendly RSs (NF-RSs) 'nudge' the optimal recommendation list, intending to prioritize low-network-cost content while also adhering to the user's preferences. This strategy aims to reduce network costs, but with the risk of dissatisfying content owners/producers by affecting the popularity of their content while also affecting the overall user experience.

The main idea in recommendation systems (RS) is the production of recommendations that will attract the user's interest. The Baseline RS (BS-RS) is the most simple yet efficient form of an RS, where recommendations are picked from a ranking system that sorts items based on past user preferences. BS-RS recommends the TOP-N contents with the highest ranking score. The list that the Baseline RS has promoted is about to be stressed by the NF-RS.

The NF-RS favors contents that are cached and can be delivered at a low network cost. In network-friendly RSs (NF-RSs), caching involves temporarily storing copies of different contents on servers placed strategically in various geographic locations. The network-friendly recommendations are included in a new list that differs from the original (Baseline) recommendations list. The divergence of these two lists signifies how each of these systems utilizes its contents and how different the recommendations may be for the user. However, the concept of *Caching* has exposed another major issue about the fairness of network-friendly recommendations (NFR). NF-RS algorithms might recommend an item that is (perceived as) slightly less interesting, but the user might still benefit for this new recommendation if this new content can be delivered with higher quality (e.g., an item cached close to the user can be streamed with fewer interruptions and often better quality). Real studies exist that demonstrate users are often happy to accept this tradeoff [1]. While NFR may improve network efficiency, it can also result in unfair treatment towards specific content, which is detrimental to content providers. In an effort to quantify this unfairness, researchers have proposed several metrics.

Specifically, the study of [2] explores several findings on fairness in network-friendly recommendation (NFR) systems, employing a diverse range of metrics, including F_{max}, F_{tv}, and F_{kl}, that capture multiple notions of (un)fairness. It claims that NFR systems are, in fact, generating injustice. It proposes approaching the construction of a *FAIR-NFR* algorithm as a linear problem to effectively address and mitigate unfairness. The main idea is to assign fairness constraints in order to control the allowed unfairness and monitor the network's performance. By using the previously discussed metrics, the researchers noticed that by tolerating a certain degree of unfairness, significant improvements can be achieved in network performance.

Now that we have presented the concept of fairness in Network friendly Recommendations we will now define the problem of "Content bubble" (lack of diversity) that this thesis describes.

1.3 Problem Statement

Although network-friendly recommendation algorithms achieve a great network-cost reduction, they also might cause the recommendations to lack diversity. This new aspect of unfairness has not been addressed by previous methods and has created the need to enable new metrics that will evaluate the variety of network-friendly recommendations.

Fairness is often categorized into Individual and Group fairness, with the former emphasizing treating *similar items* similarly and the latter focusing on treating *groups of items* similarly based on one or more protected attributes. In this particular scenario, the main focus is on the individual profile of fairness, which considers only the items one-by-one inside the recommendation list.

Various factors, fundamental to the operation of a recommendation system, can impact fairness. Cache size, number of recommendations, or probability distribution of the recommendation list are only a few to address. Changes in these factors can have a significant impact on both fairness and network cost. As already mentioned, recent work [2] has examined the behavior of the above factors under the scope of introducing fairness in order to produce network-friendly recommendations that do not diverge a lot from the baseline (original) recommendations. Maintaining high network performance contributes to a better user experience due to the fact that service latencies are reduced, leading to higher user satisfaction. However, it needs to be considered that the user is not only satisfied by the response time of the service, but one of the original goals of the RS is to surprise and delight users with relevant and novel recommendations that they would not have found or considered otherwise.

In literature, the above issue of limited and biased recommendations which may result in problems like user discontent and reduced exploration of diverse content is defined as ***Content Bubble***.In order to examine the impact of Fairness on *recommendations' diversity*, we need to employ metrics that capture the unpredictability of the recommendations and the imbalance of the distribution inside the probability demand vectors.

We will now set the targets of this thesis and analyze how we are going to examine the diversity issue.

1.4 Objectives

This thesis provides a new perspective on the fairness results discussed in the paper [2]. The paper's research has provided us with data about the functionality of network-friendly RS where they have quantified (un)fairness. The algorithms used in the research delivered results about fairness in a network-friendly environment, and they have been compared with the Baseline RS. This unfairness occurs between recommended items in recommendation lists that changed after considering network cost compared to the recommendations of baseline RS.

The purpose of this thesis is to examine another notion of fairness of the new recommendation list which is based on the diversity of the recommended items. For this reason, we will propose ***Entropy*** and ***Gini Impurity Index*** as two metrics that are suitable for this situation. Both of these new metrics will offer valuable insights into the unpredictability of the contents within the recommendation list generated by the Fair-NFRS algorithm. Our research will investigate how fairness constraints affect the diversity of recommendations to address the problem of "Content Bubble" while also ensuring network gains.

Chapter 2

Literature Review

We will examine the fundamental algorithms used by a recommendation system to generate suggestions. The primary methods of filtering are Collaborative Filtering and Content-based Filtering. In addition, we will present fundamental equations employed in Recommendation Systems and introduce two primary models that we will research: Baseline and Network Friendly Recommendations.

2.1 Filtering Algorithms in RS: Collaborative Filtering

One of the initial algorithms used in RS is collaborative filtering (CF). Collaborative Filtering suggests items by evaluating similarity measures (e.g *cosine similarity*) between users that like/dislike the same items they've already both consumed and are likely to agree also on items that one or both of them have not yet consumed.

One common scenario where collaborative filtering comes into play is during social gatherings, such as movie nights with friends. For instance, when planning a movie night with a group of friends, each person in the group has their own unique tastes and preferences when it comes to movies. Some prefer thrillers, while others lean towards comedies or drama. The group engages in a filtering process to ensure an enjoyable movie-night for everyone. People start expressing preferences about certain movies, ranking them and then a game of influence begins inside the group. In the end, a movie will be selected that satisfies the preferences of the majority while respecting the diverse tastes of all participants.

Collaborative filtering captures the preferences of a group of people and uses them with the aim of providing recommendations to another group of people with unknown preferences [3]. To achieve high efficiency in its recommendations, CF is distinguished into two main categories :

- User-based CF
- Item-based CF

2.1.1 User-based Collaborative Filtering

In this technique, the algorithm makes predictions based on the preferences of the users who are similar to the target user and aims to provide recommendations that those similar users liked/rated.

Concept of similarities in User-based CF

Two users x,y compare their similarities based on their ratings .If user y consistently rates items similar to user x, i.e., y gives item i_1 a high rating when x gives it a high rating, *and* y gives item i_2 low ratings when x gives it a low rating too, then x and y will have a high similarity weight. The concept involves suggesting an item to user x that has not tried yet, based on the high ratings given to it by several users similar to user x.

Some popular algorithms that measure similarity in collaborative filtering are Cosine similarity and Pearson similarity [4].

User-based CF Algorithms

The typical model of collaborative filtering is usually based on the User-Item Matrix where two lists: one of m users and one of n items create a matrix where each user has a list of items which has rated or about which their preferences could have been derived from their past behavior.The matrix cells store ratings of user interactions with the items. If a user has not engaged with an item, the related cell could be blank or contain a symbolic value (eg Nan).

Scenario 1

Consider the following scenario: we have a collection of people (who we will refer to as User 1, User 2,..., User N) and a collection of items (which may include movies, books, or products). Each user has submitted ratings for some of the items, resulting in the formation of a user-item matrix in which rows represent users and columns represent items. A user may not have seen a specific movie so there will be no rating from that user ("-" no rating). Lets say we have 2 users and 4 movies.

User 1 ratings : - , - , 5 , 2
User 2 ratings : 2 , 5 , 5 , 1

By observing the ratings we can see that the users' ratings are pretty similar. Suggesting movie 2 based on the similarities of the rating lists would be a good recommendation .

Based on user ratings, we calculate the cosine similarity between users, which is a similarity metric that measures the cosine of the angle between two vectors reflecting the ratings of two users; a value of 1 denotes perfect similarity, and a value of -1 denotes perfect dissimilarity.

In the context of recommendation systems, cosine similarity is often used to measure the similarity between user preferences and item features or between items themselves. In collaborative filtering-based recommendation systems, cosine similarity can be used to measure the similarity between users or items based on their interaction patterns.

- Cosine similarity is measuring the cosine angle between two users' rating vectors. Cosine similarity is a measure of similarity between two vectors in an n-dimensional space. It measures the cosine of the angle between the two vectors and provides a numerical value indicating how similar or related the vectors are.

  Formula : $\cos\_\text{sim}(u_i, v_j) = \frac{\sum_{k=1}^{n} r_{ik} \cdot r_{jk}}{\sqrt{\sum_{k=1}^{n} (r_{ik})^2} \cdot \sqrt{\sum_{k=1}^{n} (r_{jk})^2}}$
  This formula calculates the cosine similarity between two users u_i and v_j, where r_{ik} and r_{jk} are the ratings of users u_i and v_j for item k, and n is the total number of items.

- Pearson Correlation : Pearson correlation follows the pattern of cosine similarity but instead measures the extent to which two variables linearly relate with each other, where the $i \in \mathcal{I}$ summations are over the items that both the users have rated and $\bar{r}_u$ is the average rating of the co-rated items of the u-th user.
  [3]. Formula: Pearson Correlation$(u, v) = \frac{\sum_{i \in I} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{vi} - \bar{r}_v)^2}}$
  In the formula u and v represent the users and r_{ui} is the rating of user u and item i. It also includes the mean rating of user u over all items by u $\hat{r}_u$.

2.1.2 Item-based collaborative Filtering

Another similar model is "Item-based" collaborative filtering which is also used in marketing platforms such as Amazon [5]. In this scenario, item-based collaborative filtering identifies similarities between items based on user ratings and recommends new items to users based on the similarity of their ratings to items they have already rated positively.

Using this technique, a matrix (item-item matrix) of similar items is produced based on items that users tend to buy together. The quickness of this computation varies on the number of items the user has bought or rated.

Item-based CF Algorithm

The same logic as the User-based algorithms but instead they change some variables that are relevant to items more than users [3].

- Pearson correlation: Set of users $u \in U$ who have rated items i and j and the rating r changes as the rating of user u in item i as r_{ui}. Now $\hat{r}_i$ represents the average rating of the i-th item by the users. Formula: Pearson Correlation$(i, j) = \frac{\sum_{u \in U}(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{u \in U}(r_{ui} - \bar{r}_u)^2}\sqrt{\sum_{u \in U}(r_{vi} - \bar{r}_v)^2}}$

2.1.3 Cold Start Problem in Collaborative Filtering

One of the basic problems in Collaborative Filtering is "user/item cold start " which happens when a new user registers and has no previous preferences or when a new item is listed and there is limited interaction with users of the system, but this is a problem which is discussed in other research papers . [6]

After analyzing the structure and capabilities, but also the problems of Collaborative Filtering, we will now continue researching filtering algorithms.

2.2 Filtering Algorithms in RS: Content-Based Filtering

Another method of filtering is based on content variations where the recommendation is not only based on similarity by rating but focuses more on the information that can be extracted from the items [7]. Content-based filtering utilizes the User-Item matrix differently due to the fact that it emphasizes more on the content features of the item and not only the preferences between similar users.

Content-based filtering uses the features or characteristics of items to create suggestions that resemble something the user has previously engaged with or liked. The user-item matrix displays ratings assigned by users to items, facilitating the creation of user profiles and the computation of item similarities based on their features.

This filtering method seems to handle the cold-start user problem better than collaborative filtering because it emphasizes on the attributes of every item and does not require extra information about the user interaction data. The problem with content-based filtering is that it focuses more on the features and less on the user's ratings which can lead to less serendipity and can cause the recommendations to lack variety.

Content-Based Filtering Algorithms

In the context of content-based filtering method the above similarity algorithms could also be suitable, but in some cases, they represent items as sets of features (words, etc). In content-based filtering, *cosine similarity* is often used to calculate the similarity between item feature vectors, where each dimension represents a feature or attribute of the item. In order to identify the items that are most similar to each other in content-based filtering, these similarity approaches are employed to measure the similarity of items based on their characteristics or content features. The selection of the similarity technique depends on several criteria, including the type of data, item attributes, and the particular objectives of the recommendation system. A very common algorithm that provides similarity measurement is *Jaccard Similarity* [8].

In a content-based recommendation system, the cosine similarity between the feature vectors representing items and user preferences can be calculated to identify items that are most similar to the user's interests.

Scenario 2

Assume we have a dataset of films, each of which is classified into a number of genres. Our goal with content-based filtering using Jaccard similarity is to suggest movies to consumers based on the genre similarity between the films in the dataset and the films they have previously enjoyed.
Let's take an example where a user of the Netflix platform has indicated that they like dramas, thriller, and crime. We calculate the similarity between this set of genres and the genres of every movie in the dataset using Jaccard similarity. The user-liked genres and each movie's genre are the two sets that the Jaccard similarity measure overlaps between.

This method ignores other elements like user ratings and preferences and instead depends only on the substance (genres) of the films. It is especially helpful when we know a lot about an item's content but little to nothing about the preferences or interactions of the user.

– Jaccard Similarity : Dividing the slice of two sets by the combination of these two. Jaccard Similarity$(A, B) = \frac{|A \cap B|}{|A \cup B|}$ In this formula $|A \cap B|$ is the size of the intersection of features between items A and B. $|A \cup B|$ size of union between A,B.
  Then :
  $d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$
  The closer it gets to 1 the higher the similarity of these two sets of A, B

Now that we have analyzed Content-based filtering and examined some of the similarity methods that are used, we will now proceed to the algorithm that is mostly utilized due to the fact that it can combine both Collaborative and Content-Based Filtering.

2.3 Filtering Algorithms in RS: Hybrid Filtering

Hybrid Filtering is one of the most commonly used filtering algorithms in recommendation systems. This method integrates many filtering algorithms to generate results with increased accuracy. Netflix uses hybrid filtering to generate suggestions by analyzing the browsing and viewing habits of similar users and suggesting content with similar features which is highly-rated. [9].

With the hope of satisfying the flaws of the Collaborative and Content–Based filtering methods, Hybrid filtering is widely used by many content platforms, such as Youtube, Netflix, Spotify, etc. Hybrid filtering might not only combine CF(collaborative) and CB (content-based) filters but could also include Demographic Filtering and Knowledge-Based filtering.

Other types of filtering algorithms

Demographic filtering

Demographic filtering utilizes data such as age, gender, and other relevant factors to deliver customized content that is employed in educational platforms and etc. Although it seems to deal better with the problem of a *cold start* user due to the fact that it does not require any kind of ratings, DF seems sometimes to exceed the limits in personal data usage and for that reason, it is not easily applicable in its original form.

Knowledge-based

Knowledge-based filtering uses knowledge about users and items to reason about which items meet users' requirements [10]. A standard hybrid filter algorithm could be described with different methods by assigning weights to each recommendation as it is examined in this research paper [11].

We will now present some of the most basic but also some advanced recommendation algorithms used in RS.

2.4 Recommendation Algorithms

Baseline models are models used as a starting point to recommend data in a simple yet effective way. Some algorithms used in the baseline recommendation system are :

Matrix Factorization

Matrix Factorization involves breaking down a user-item interaction matrix into the product of two lower-dimensional matrices. This feature makes it suitable for Collaborative filtering.

- The original matrix R has users-items as rows and columns and the entries are the ratings from the users to the items. Here the rating is represented as R_{ui} where u is user and i is item.
- Then the matrix R is decomposed into the product of U and V: $R \approx U \cdot V^T$
- U is the user matrix and every row u_i corresponds to the latent factors associated with user i. V is the item matrix where each row v_j corresponds to latent factors associated with item j
- The rating of the user u and item i is the product of $r_{ui} = u_i \cdot v_i$

Matrix factorization may face difficulties with new user entries which makes it struggle with the problem of ***Cold Start***. Its efficiency may drop with highly sparse matrices.

Global (Average) Baseline

One of the most simple techniques in baseline recommendations is the Global Average. It assumes all users and items have the same average rating. This approach provides a basic prediction by considering the average rating and stands for the standard choice.

Formula:
$S_{ui} = \mu$
Where μ is the average rating

$R \approx U \cdot V^T$	U user matrix, V is the item matrix
$r_{ui} = u_i \cdot v_i$	The rating of the user u and item i
$S_{ui} = \mu$	Overall Mean

Table 2.1: BS RS Models

- User-Baseline $S_{ui} = \mu_u$ Where μ_u is an estimate of the u user average rating.
- $S_{ui} = \mu_i$ Where μ_i is an estimate of the i item average rating.

A more generic formula for the Global average is also :
$S_{ui} = \mu + b_u + b_i$
where b_u and b_i are the user and item bias, which encapsulates the scenario where a user rates high or low on average and an item is rated higher or lower than average. The above formula is sometimes referred to as personalized mean baseline [12].

Top-N Recommendation Algorithm

Top-N recommendation algorithms recommend the top-N highest-rated items to users. These recommendations are based on aggregate statistics and are typically ranked according to their popularity or rating scores [3].

e.g *The algorithm in a social media feed might prioritize posts according to user engagement indicators, such as likes and shares, and show the top-k content at the top of the stream.*

- User-based top-N recommendation algorithms firstly identify the most similar users (nearest neighbors) to the active user using the Pearson correlation, in which each user is treated as a vector in the m-dimensional item space and the similarities between the active user and other users are computed between the vectors.
- Item-based top-N recommendation algorithms which help to address the scalability problem of the User-based top-N

ML Recommendation Algorithms

There are some Machine Learning algorithms about recommendations which are discussed in the paper [13]. Specifically, it utilizes classifiers from ML such as SVM (Support-Vector Machines), Logistic Regression, and K-nearest neighbors. These will predict the nearest categories from the 'News Category Data' where among these categories the most common sentence will be recommended to a user.

Q-Learning

Q-learning is a reinforcement learning technique that has been applied to RS in recent years. Some researchers have deployed Q-learning to optimize long-term user engagement in feed-streaming scenarios such as mobile apps. Recommenders focus only on instant metrics but Q-Learning will help to emphasize on long-term user engagement by including factors such as dwell time and revisit frequency [14]. As it can be seen Q-learning emphasizes on personalization and in RS Q-values might represent the expected utility or user satisfaction associated with recommending a particular item in a given user context.

Deep Neural Network in Recommendations

Deep neural networks are also utilized in the field of recommendation systems with the aim of enhancing their capabilities. This model integrates a collaborative filtering recommendation method with deep learning technology, including two components. The initial step involves employing a feature representation technique that relies on a quadric polynomial regression model.

The proposed method enhances the accuracy of obtaining latent features by enhancing the conventional matrix factorization approach. Subsequently, these latent characteristics are included as the input data for the deep neural network model, which is the second component of the suggested approach and is employed to predict the rating scores [15]. In the above paper, the whole model of DNN and also other methods that are used such as matrix factorization using SVD, are explained in depth.

2.4.1 Filtering algorithms in famous platforms

Filtering methods in Netflix

Netflix uses a more complex recommendation system that handles a large amount of data and many on-demand users' requests. To satisfy the customers' needs Netflix utilizes collaborative filtering to analyze users' preferences and viewing history but also provides content that follows a viewing pattern of users with similar interests [16].

Moreover, it collects results from an item-based collaborative filtering method that concentrates on similarities between items. Netflix builds a user's profile based on their past preferences, which is always evolving as the user interacts with the platform. According to a paper, a typical Netflix user loses interest after approximately 60 to 90 seconds of browsing, having examined around 10 to 20 items, with perhaps 3 in detail, on one or two screens. The user either finds something tempting and interesting to watch or closes the streaming platform and continues somewhere else. Netflix's recommender challenge is to ensure that each user will find something engaging to watch and that will capture their interest. [17].

Data personalization is a difficult goal which is crucial for the platform's success and is the key element to increasing user engagement. Another method of personalized data is user feedback where the platform in its application uses thumbs up/down indicators in order to collect information about movie preferences of the user.

Netflix utilizing filtering methods

Some of the most famous features of Netflix's platform are crucial for keeping the user engaged and creating the best possible user experience. Some research papers analyze and present the user interface of Netflix and describe what recommendation techniques and filters are used in order to optimize their proposals [18].

- Personalized video Ranker (PVR) is one of the recommendation algorithms used in Netflix which sorts the catalog of videos according to user's preferences.
- The Top-N Video ranker is a mechanism that seeks to identify the most essential customized recommendations from the catalog for each member and display them in the Top choices line.
- Trending now is a method that utilizes a content's popularity and predicts that the users will also like it and recommend it.
- Many more algorithms like *Continue watching* are constantly trying to enhance the services of Netflix which ranks the unfinished videos by estimating which of them the user is more likely to click on. In order to help the user, the application provides information about where and when the video has stopped.
- Video-Video similarity collects personalized data based on the scenario of -*before you watched*-. This algorithm is also called *sims* which initially is an unpersonalized algorithm that computes a ranked list with similar videos for each and every video in the catalog.

Tiktok's Recommendation Techniques

Several factors contribute to the relevance and interest of material before it is recommended to a viewer. Each recommendation system picks videos from a large pool of potentially interesting material and ranks them by how likely it is that the user will be interested in each one. These predictions are also affected by how people on TikTok, who seem to share the same interests, connect with each other. User A might like videos 1, 2, and 3 if User B likes videos 1, 2, 3, 4 and 5, as an example. The recommendation system might guess that User A will also like videos 4 and 5. In other words, each feed is different. People may see the same videos, but your ***For You, Following, Friends, and LIVE feeds*** are based explicitly on the user's past preferences [19].

TikTok's "For You" feed features a blend of videos from established online personalities and rising creators, prioritizing top-notch creative content based on viewership metrics, and supporting new bloggers by showcasing their videos to consumers. What sets it apart is that anyone can potentially achieve fame on the platform. TikTok's recommendation algorithm suggests videos to users based on shared interests or characteristics with video creators, facilitating the rapid spread of high-quality creative content. The TikTok recommendation algorithm does not prioritize a video blogger's followers or popularity. The algorithm incorporates the user's uploaded video content and the categories of videos liked by the user, in addition to the candidate video title, audio, and tags.

Final recommendations are generated by traditional ***collaborative filtering (CF)*** and ***content-based*** recommendation techniques to personalize the user experience. TikTok's recommendation algorithm effectively suggests movies based on user interests and introduces them to new topics, satisfying their desire for novelty and unexpected discoveries [20].

2.5 Key concepts and equations

The main goal of network-friendly RS algorithms is the following: (i) take the suggestions and similarity scores of state-of-the-art algorithms, like the ones described, as *input* and (ii) 'nudge' them towards items that have a lower network cost, when possible. Since there's a lot of complexity in the underlying baseline RS algorithms, we describe here an abstract RS and (resulting) user behavior model, that attempts to capture some of the salient features of such recommenders. These definitions are utilized in the implementation of this thesis and some can be found on tables 2.5 and 2.1.

Recommendation systems focus on user preferences and try to enhance the recommendations that are provided. First of all, we consider a user that consumes one or many contents during a session which is selected by a catalog $\mathcal{K}(|\mathcal{K}| = K)$ of cardinality K. User requests content in two ways :

1. following an external link or typing the desired content through a search bar perhaps

2. following the recommendations provided by the RS service

Let i be the content (item) to be consumed. Content i belongs to the catalog K. While the user selects content i, a list of N new contents will be recommended for future selection. There we have two scenarios:

- the user follows the recommended option with some fixed probability $a \in [0, 1]$ and picks uniformly among the N contents

- the recommended item i is being ignored with probability $1 - a$

Defining the demand p_i for content i as the fraction of all requests without considering it is direct from the user or comes from the recommendation system. We denote p as $\mathbf{p} = [p_1, \ldots, p_K]$ where p is the vector representing the distribution of overall demand for all contents.

Content Relation in Matrix U

The values of u_{ij} populate the square $\mathcal{K}$x$\mathcal{K}$ matrix named as $\mathcal{U}$. u_{ij} values also correspond to the cosine similarity between content i and j which is a method used in Collaborative filtering.

Score of recommendation u

To evaluate the efficiency of a recommendation, we utilize the recommendation score u_{ij} for every pair of contents $i, j \in K$ which expresses how good the recommendation for content j is, after consuming the content i from catalog K. This score indicates the similarity between these two contents, which is a logic derived from *item-item* Collaborative filtering method. In general, if the score $u_{ij} \in [0, 1]$ has a 'zero' value, it stands for a very bad recommendation score, and if it is 'one' this means that the recommendation was very successful.

Baseline RS (BS-RS) generates the recommendation scores u_{ij} and is used in production by the content/service provider. After a user has consumed content i, the BS-RS recommends to the user a list R_i^{BS} i that contains the N contents with the highest recommendation score values.

Denoting the probability of the next recommendation r

A very important variable is r_{ij} which measures the probability of content j to be recommended after content i. These probabilities define a square $\mathcal{K}$x$\mathcal{K}$ recommendation matrix R over which we optimize.

$\mathcal{K}$	set of cardinality
N	number of recommendations
a	probability a user to follow recommended item i
p_i	probability for the demand of content i
$\boldsymbol{p}$	the vector with the distribution of total demand for all contents
u	score of recommendation
U	content relation Matrix
r_{ij}	the probability of content j to be recommended after content i.

Now that we have presented all the basic equations that are included in a recommendation system and will be utilized in this thesis, we will now introduce the concept of caching. Caching will enhance the performance of the recommendation system and will play a crucial role in network-friendly recommendations.

2.6 Caching in RS

On every single platform that provides streaming services, like Spotify and Netflix recommendation systems play a crucial role in their functionality. Traditionally, a recommender will aim to provide recommendations that are based on the user's interests in order to create a unique user experience while using a service [21]. Recommendation systems focus on individual user preferences. The recommended material should be highly attractive to the user in order to encourage further content consumption. The results of the above filtering algorithms (CF, CB, and HF) require many resources to be produced and that leads to a high computational load.

In the recent years, the concept of ***Caching*** has been introduced to Recommendation Systems in order to improve their performance and efficiency.

Caching is utilized in RS :

- By storing user profiles in memory (or a distributed cache), the system can quickly retrieve relevant information about users during recommendation generation.
- By precomputing and caching item representations such as feature vectors, the system can avoid repeatedly processing item data .
- By storing similarity matrices in memory or a distributed cache, the system can quickly access similarities that have already been evaluated between items.

Generating recommendations may be resource-intensive, particularly in systems with extensive datasets or complicated recommendation algorithms. Caching enables the system to save and reuse suggestions, hence decreasing the computational workload. Caching enhances the scalability of recommendation systems by minimizing the calculations needed to create suggestions, enabling the system to accommodate more users and requests [22]. The content to be cached is selected to ensure that the cache serves the largest aggregate demand of all users. [23].

Cache Hit Rate

In recommendation systems, a very important metric is the Cache hit rate. It is used to measure the effectiveness of the caching technique used in recommendations and reflects how well the caching system is able to retrieve the recommendations from the cache [2].

A high value of the variable CHR indicates that a significant part of the recommendations can be retrieved from the cache leading to a reduction in computational load which is beneficial for the network's health. Providing recommendations that are pre-computed and already stored in Cache enhances also the system's response time.

- Formula of Cache Hit Rate (CHR) :

  $CHR = \sum_{i \in \mathcal{C}} p_i$

  Where p_i is the probability demand for content in Cache $\mathcal{C}$

2.6.1 Caching in RS of famous platforms

The success of mobile Internet applications such as TikTok, Instagram, and YouTube relies on extensive video libraries and, crucially, on the efficient architecture of the caching mechanism. Furthermore, short video platforms with massive user databases consume a significant amount of Internet bandwidth. Additionally, they are time-sensitive, particularly for brief video services such as TikTok. TikTok presently has over 150 million active daily users in China, with the average size of an uploaded video file being 1.96 MB. If each user uploads a 1.96 MB video daily, the total disk space needed to store all the videos is around 294 TB. Hence, dynamic and efficient caching is essential for such platforms. Bandwidth cost and end-to-end latency are significant concerns for TikTok. Ensuring Quality of Experience is undoubtedly the most significant problem it encounters. Edge caching can decrease backhaul bandwidth utilization and minimize delay, which is crucial for capacity planning and improving Quality of experience (QoE) [24].

Edge caching in TikTok

TikTok probably utilizes a distributed network of edge servers strategically positioned globally. Edge servers retain cached versions of popular or frequently accessed videos. TikTok's technology directs user video requests to the closest edge server containing a cached video copy, minimizing latency and accelerating delivery.

TikTok might collaborate with different Content Delivery Network (CDN) providers to enhance video distribution. Content Delivery Networks (CDNs) distribute content regionally to offer videos from servers located closer to consumers, lowering latency and enhancing streaming performance. Content Delivery Networks (CDNs) frequently incorporate caching systems to retain duplicates of videos at various points around their network.

In fact, 80 % of the streaming hours on Netflix is due to the recommendation system that is used and 30 % of YouTube's overall views are from the recommended videos [18], [25]. The personalized user data is cached in order to create a faster service and recommendations are "nudged" towards interesting content with low-access cost [22]. Furthermore, caching techniques contribute to scalability challenges, especially in large databases where by frequently caching users' data, recommendations systems reduce the back-and-forth data exchange from backend storage which endorses a healthy and steady network functionality.

After analyzing the concept of caching, we will now proceed to the recommendation model that utilizes caching. Network-friendly recommendation systems are based on the logic of Caching while equipping the algorithms of Baseline RS but also considering network performance.

2.7 Inducting a Network Friendly profile in RS

Caching techniques in Recommendation systems have opened a new world in content distribution but have also introduced new challenges to the scientific community. As stated before most of the requests are mostly driven by recommendation systems and that requires making RSs favor locally cached content so that to reduce latency and create a better user experience while reducing network costs to benefit also content providers.

The main idea is to promote high-quality recommendations meaning to select recommendations that are more efficient network-wise. To achieve that kind of efficiency, content providers have expanded the original logic of **Content Delivery Networks** (CDNS) and embedded it into their own Network by installing small data centers inside the network operator. Now content that is frequently accessed(such as Users' profiles) will be cached strategically and this will reduce the computational overload that servers go through in order to please and handle successfully every request that is sent to them.

By caching strategically, the recommendations will still have a good quality and the action will require a smaller network cost. Furthermore, if the network is not stressed, it will not face latency problems and this is extremely important in streaming platforms such as Netflix where the quality of the streamed video must always be the best possible so that it creates a smoother and more engaging user experience.

NF-RS Setup

Following the same logic as the Baseline Recommendations, NF-RS has the same setup but with more expertise around Networks.
A Network Friendly Recommendation System is a type of recommendation system that considers various network conditions including delivery costs , cached contents, and more [26] [27].
The NF-RS provides an alternative list of recommendations R_i^{NF} which may

$q_i^{max} = \sum_{j=1}^{\mathcal{K}} r_{ij}^{base} \cdot u_{ij}$	Baseline RS Policy
$\sum_{j=1}^{\mathcal{K}} r_{ij} \cdot u_{ij} \geq q \cdot q_i^{max}$	NF RS Policy
$CHR = \sum_{i \in \mathcal{C}} p_i$	Cache Hit Rate

Table 2.2: QoR (BS-NF) and CHR

be identical to the baseline recommendation list R_i^{BS} and even may share some common items or none in some cases.
Typically, NF-RS recommendations :

1. include more suggestions for contents that can be delivered in a network-friendly manner, such as cached contents

2. while striving to maintain the quality of recommendations (QoR) by suggesting contents with relatively high scores u_{ij}.

The main effect of NF-RS is that it influences the demand p. If a particular content i is frequently selected then the demand of p_i will be increased.
The Demand p is described in NF-RS as p^{NF} and in Baseline Recommendation Systems (BS-RS) as p^{BS}.

Network friendly subset $\mathcal{C}$

The subset $\mathcal{C} \subset \mathcal{K}$ describes a subset of the content catalog which is recommended with low networking cost. The cost for delivering contents that belong in C is established to be 0. Otherwise, the cost for delivering some other notion of contents of the catalog is 1.

NF-RS QoR

As stated in the Baseline Recommendations the system achieves a quality of content i equal to : $q_i^{max} = \sum_{j=1}^{\mathcal{K}} r_{ij}^{base} \cdot u_{ij}$ Where q_i^{max} symbolizes the maximum quality from the baseline recommendations. After the user consumes the content i, the baseline recommendation system recommends a list of R_i^{BS} with a size of N contents that have recommendations with the highest score values of u_{ij}. The baseline RS, for every content $i \in \mathcal{K}$, will always recommend the N items with the highest u_{ij} scores [21].

A multifaceted strategy has been taken to include a network-friendly profile into Recommendation Systems. One of these approaches involves the implementation of a new policy that considers access cost and ensures a certain degree of Quality of recommendations (QOR). This concept has been proposed in paper [21] which investigates network-friendly recommendations for optimizing long Viewing Sessions.

- The policy established for this paper about NF-RS is:

  $\sum_{j=1}^{\mathcal{K}} r_{ij} \cdot u_{ij} \geq q \cdot q_i^{max}$
  Where the q here describes a notion of q_i^{max} for every content i which follows a set of K inequality constraints.

The tuning parameter q is a critical factor in the RS that determines the percentage of the maximum quality (q_i^{max}) offered. Significantly, as q approaches 0, QoR diminishes, and the RS makes recommendations solely based on access cost, presenting an opportunity for substantial network gains. Conversely, when q approaches 1, the RS transforms into R^{BS}, indicating a highly constrained optimization problem where the RS cannot enhance network access cost.

In this paper, they try to implement an Absorbing Markov Chain in order to solve the problem and optimize the network-friendly recommendations [21].

2.8 Proposed NF-RS algorithms

The paper [2] presents some Network-friendly recommendation algorithms that are explained more briefly inside it.

2.8.1 Greedy NF-RS

The Greedy NF-RS strategy aims to populate each recommendation list (R_i) with numerous cached contents while ensuring that a minimum Quality of Recommendation (QoR) threshold q is not breached. Its objective is to maximize the Content Hit Rate (CHR) by treating each request in isolation, without considering long-term performance implications. [2] [28]

- Other algorithms used and analyzed inside the paper are : **Multi-step NFRS** [29] and **CaBaRet** [26] [30]

Now that we have presented Network-friendly recommendations we will focus on the problem that is created by this type of RS. Fairness is one of the key elements in this thesis and will be analyzed in the following sections.

2.9 Fairness Definitions

2.9.1 Group Fairness

When examining group fairness criteria, research typically involves categorizing subjects into distinct groups using a specific grouping method, with the most common approach being the segmentation of users based on their sensitive attributes. The determination of which features qualify as sensitive is subjective, as individuals may have varying definitions. In general, sensitive features refer to inherent characteristics that individuals do not have control over, such as gender, racial background, and age. The core of sensitive features resides in characteristics that are not influenced by individual preference. [31].

Group fairness requires that machine learning systems treat protected groups in a manner that is comparable to advantaged groups [32]. When discussing gender discrimination in a recruiting decision-making system, candidates are initially categorized into groups according to their gender. Fairness evaluation entails the process of measuring the system's disparate treatment towards different groups, taking into account elements such as income or hiring rates. Assessing and rectifying inequalities within the system necessitates a thorough examination of fairness [31].
An example of (a concept though) Group fairness is presented in the paper of [33] where the researchers analyze the problem of recommending a package of items to a group of users while ensuring that every group member is satisfied by a sufficient number of items in the package.

2.9.2 Individual Fairness

Individual fairness, as opposed to group fairness, requires treating comparable individuals in the same way. In order to maintain fairness for each individual, it is crucial to evaluate whether two individuals may be considered similar based on a specific measurement, such as the distance between their user representations. One can determine the similarity between individuals by examining specific sensitive attributes or a combination of them. For example, individuals who fall within the same age range and gender can be considered similar. Alternatively, hidden characteristics can be utilized to determine user similarity. In this approach, each user is represented by a latent vector that is created from their profiles or interaction data. The similarity between users is then established by comparing the similarity of their vectors [31].

To illustrate this, consider the scenario of addressing unfairness in a hiring decision-making system. Each candidate can be represented as a vector, and the requirement for individual fairness entails that candidates with closely aligned representations receive comparable salaries.

It is important to emphasize that group fairness and individual fairness are distinct concepts. For instance, if similar individuals are grouped together, individual fairness underscores fair treatment within the same group, whereas group fairness emphasizes fairness across different groups at an aggregated level. An approach that is individually fair may not necessarily be fair at the group level, as fairness within a group does not guarantee parity among different groups. Conversely, a method that is fair at the group level may not ensure individual fairness, as it is plausible for fair treatment at the group-aggregated level to conceal disparities among individuals within the same group. Consequently, neither group fairness nor individual fairness encompasses the other, necessitating careful consideration of both concepts.

Example of Connection between individual fairness and group fairness

Statistical parity refers to the condition where the demographic composition of individuals receiving positive (or negative) classifications mirrors the overall demographics of the entire population. Despite its appeal in promoting equalized outcomes, we highlight its insufficiency as a measure of fairness through various examples where statistical parity is maintained, yet individual outcomes are evidently unfair [31].

2.9.3 P-fairness in RS

A system that mandates P-fairness indicates that content providers are accountable for prioritizing fairness. P-fairness is significant in circumstances where prioritizing market diversity and limiting monopolistic control are important. For example, in the digital craft marketplace Etsy, the system may strive to ensure that new participants receive a fair amount of recommendations, even if they have less customers compared to experienced sellers. While not legally required, this type of justice is deeply ingrained in the platform's economic strategy [34].

Producers in P-fairness are passive, awaiting users to approach the system and request recommendations, as opposed to actively seeking out opportunities. For example, in employment cases, we may desire that jobs at minority-owned businesses be recommended to highly qualified candidates at a rate comparable to jobs at other types of businesses. Recognizing and acting upon such opportunities may be infrequent, requiring a unique approach to maintain fairness without excessively sacrificing personalization.

The research on diversity-aware recommendation aims to enhance both accuracy and diversity in recommendation lists. These techniques can be modified to achieve P-fairness recommendation by considering items from the protected group as a separate category. It is important to emphasize that attaining a diverse list does not ensure fair recommendation results for all providers in the set.

Individual P-fairness necessitates the use of a dynamic model to effectively manage recommendation opportunities. Analogous to the process of online bidding for display advertising, a system that has predetermined budgets for providers might distribute resources via a second-price auction method. Ensuring individual P-fairness in this scenario entails granting the protected group identical purchasing power as the non-protected group, but according to individualized processes. The design of the bidding agent could show different levels of complexity depending on the information that is accessible. A sensible approach to this design would prioritize factors such as budget,etc [35].

After explaining the definitions of Fairness we will now illustrate the problem of NF-RS being less fair in comparison with the BS-RS.

2.10 Analyzing the problem of (un)Fairness

As stated before,NF-RS increase the performance of the network by using caching techniques but it comes with the cost of being less fair towards some other contents in comparison with the Baseline Recommendation system. This scenario of Fairness is going to be studied for the first time in this paper [2].

The network-friendly recommendation systems environment consists of 3 main entities: The network, the users or consumers who will use it and the producers/content providers. Many other papers have proposed solutions that emphasize on optimizing the NF-RS in order to benefit the network and focus entirely on user satisfaction. Except from the user, there could also be benefits for the content provider but most papers link it to the perspective of user satisfaction. In the context of NF-RS , the goal is to strike a balance between optimizing network performance and ensuring fairness among recommended items. The fairness aspect attracts the interest of researchers due to the fact that in network-friendly recommendation systems, some contents may be prioritized in order to enhance network quality but cause uneven exposure or promotion to some items. This situation is characterized as the state of Unfairness inside an NF-RS.

The problem of fairness could affect both users and content providers. In order to achieve the best possible network gains the algorithms of NFRS may change a little bit the optimal recommendations list provided to users which could lead to a lower-quality user experience. Moreover, network-friendly recommendations could also affect the popularity of some content by changing their exposure which may not be beneficial for some content providers. There could be some tolerance for unfairness in NFR coming from both sides. Content providers may tolerate certain unfairness while in network congestion in order to protect the service experience of their clients by delivering low-quality material.

Other scientists have also conducted research on Fairness but **not** in Network Friendly Recommendation Systems. They analyzed fairness in basic recommendation systems that do not consider network performance. This approach is often mentioned as Fairness in rankings.

2.11 Fairness in Rankings

The scientific community is very interested in fairness of recommendations and many papers have experimented with this subject. Although there are many proposals about metrics used in recommendations that measure fairness, most of them do not engage with network-friendly RS. This gap is yet to be filled, but many of the algorithms and ideas that were used in previous work can also be applied in systems that take into consideration network benefits.

The concept of fairness in RS is sometimes approached as fairness in rankings due to the fact that rankings are a crucial part of a recommender. Fairness in rankings takes into consideration the theory of Item and Group fairness and based on this, many algorithms are applied to satisfy the user's requests and also ensure the best possible treatment between the ranked items. This treatment is also translated as the *exposure* that every item is given [36].

Statistical Parity

The term statistical parity is also referred sometimes as demographic parity and it is often used to ensure fair and unbiased recommendations in a system across many demographic groups . Statistical parity is a concept that encourages group fairness. This measure is discussed in the paper where it is used to compute the difference in distributions of many groups with different prefixes of the ranking [37].

A multi-sided fairness problem based on exposure

A very challenging problem is to guarantee fairness towards both the consumer and the provider. In the paper of [38] they analyze the problem of dynamic pricing policy in the hotel industry, Specifically, they proposed a dynamic pricing policy that exploits a well-known game-theoretic fair solution concept, namely the Owen values, in order to compute fair prices. The whole concept of exposure is based on customer and provider fairness.

Fair Top-k Ranking

Top-K is a very common algorithm in recommendation systems and it can also be implemented to favor fairness. Especially in the paper [39] they propose a top-k ranking system but with fairness . They take into consideration the relevance between items and place them in hierarchy from less relevant to more relevant item.

This paper introduces an algorithm called FA*IR, which aims to provide a top-k ranking that maximizes utility and ensures ranked group fairness. It assumes the presence of a sufficient number of protected candidates. Ranked group fairness is determined by evaluating the extent to which certain groups are underrepresented in a given ranking. This evaluation involves examining protected groups and applying a statistical test to measure the proportion of protected candidates in various parts of the ranking.

In Chapter 2 we presented all the basic definitions and concepts that the bibliography provides about NF and BS Recommendation systems. After considering all the above information, we will now dive deeper into the problem that this thesis aims to research. In Chapter 3, we will present the only "Related Work" that has to do with fairness in NF-RS which is included in the research paper [2] that is the basis of our thesis, as it is the only paper to fully study the above problem.

Chapter 3

Reducing the Intrusiveness of Network-Friedly RSs

In this Chapter, we will analyze the paper that this thesis is based on. Researchers utilize some NFR algorithms (eg Multi-step NFRS) and based on their results, they will try to measure (un)Fairness using three metrics. Furthermore, they will implement a *Fair-NFRS* algorithm that considers the aspect of Fairness in NF-RS.

3.1 Fairness in NFR

This thesis is clearly based on the research paper [2] which is the first ever paper to study fairness in network-friendly recommendation systems. Since this paper is one of its kind the related work for this thesis is only the above paper. Fairness and network benefits are being extensively analyzed in this work and many observations have been made based on the results of the data that was tested.

This paper shifts the attention to the side effect of the network-friendly recommendation systems which is referred to as *Unfairness.*This kind of systems aim to boost network performance by suggesting cached content that can be efficiently delivered. The cost of this process is to surface fairness problems between the contents and of course create heated debates between content providers/producers.

Approaching the problem of fairness

As discussed in depth in the chapter of Theoretical Background this paper presents the difficulties that occur with fairness in NFRS and what happens during the process of applying a network-friendly profile to recommendations that affect the users' experience.

To face this kind of difficulties the researchers of the paper have applied a variety of metrics to capture the unfairness that is created and analyze it. They delve into the behavior of the network gain that the NFR achieves and how it alters as the phenomenon of unfairness escalates. The results of the experiments indicate that the Fair-NFRS can achieve high network gains with not that much unfairness.

In general, this paper aims to characterize fairness by using some metrics and quantify the unfairness. Furthermore, it examines the trade-off between fairness and network gain and proves that the problem of optimal Fair-NFRS can be defined as a linear program.

How community handles fairness

The paper also includes sets of figures that present the behavior of (un)fairness in relation with the fairness constraints that were used for every fairness function. Plotting the CDF of the values of the fairness metrics F, network gains, and other variables they reach into many useful conclusions about fairness which they deliberately analyze inside the paper. They define network gain as the increase in the cache hit rate (stated before) achieved by the network-friendly recommendation system (formula in :2.2).

Inside the paper, there is a big analysis of the relation between many system parameters and fairness. After plotting and processing the results of the NFRS algorithms with the fairness functions, many observations are made about fairness and network gain. The differences are between the baseline RS and the network-friendly RS, where the first is considered the standard and more fair RS.

We will now analyze the Fairness Metrics that are used by the researchers in order to quantify (un)fairness.

3.2 Fairness Metrics

Doing an analysis on the fairness algorithms, breaking every function down, and analyzing how it works and how they affect fairness.

3.2.1 MAX

- $F_{\max} = \max_{i \in \mathcal{K}} \left| p_i^{NF} - p_i^{BS} \right|$

Description

The function describes the maximum difference between the two distributions of the network-friendly and baseline recommendations. This maximum filter creates a very strict environment for every item since no demand difference will be more than F_{max}. This algorithm is also used as the worst-case scenario.

3.2.2 Total variation

- $F_{tv} = \frac{1}{2} \cdot \sum_{i \in \mathcal{K}} \left| p_i^{NF} - p_i^{BS} \right|$

Description

Total variation distance describes the absolute average difference between two distributions, BS and NF. In this context, a lower total variation distance signifies a closer alignment in the probability distributions of the above, indicating a more equitable recommendation system. In general, TV can express the average change in content demand.

By measuring the magnitude of distributional discrepancies, the total variation distance provides a quantitative measure for assessing the fairness of a recommendation algorithm.

This formula is a lot smoother than F_{max} due to the fact that F_{tv} can absorb the big differences of some content demands by computing smaller ones from other demands.

3.2.3 Kullback-leibler divergence

- $F_{kl} = \sum_{i \in \mathcal{K}} p_i^{BS} \cdot \log \left(\frac{p_i^{BS}}{p_i^{NF}} \right)$

Description

Kullback-Leibler (KL) Divergence plays a crucial role in recommendation systems, particularly when evaluating fairness in recommendations. It helps measure the disparity between predicted preferences or recommendation scores and the actual user preferences, making it a valuable metric for quantifying the difference between two probability distributions.

F_{kl} has a more complex formula than the other two and emphasizes on the p_{BS}. Due to the logarithmic component f_{kl} for lower p_{BS} values, if the demand increases, then the result of the f_{kl} divergence has a higher increase.

For all the metrics, there has been a normalization by the researchers, which leads to a range of values $\in [0, 1]$

3.3 Inducting Fairness Constraints (optimal FAIR-NFR)

For every item i $i \in \mathcal{K}$ both of the three functions capture the resulting unfairness as $F = f\left(\mathbf{p^{BS}}, \mathbf{p^{NF}}\right)$. As the paper includes, their main objective is to formulate the problem of designing optimal network-friendly recommendation systems considering fairness. To solve this problem they will model it as a linear program.

In order to solve this problem, they introduced the concept of ***constraints***. Especially for fairness , they created a threshold c_f as $F\left(\mathbf{p^{BS}}, \mathbf{p^{NF}}\right) \leq c_f$ which indicates the maximum allowed unfairness (F is one of the three functions). Moreover, another constraint as we mentioned above, is Quality of recommendations constraint $q \in [0.5, 0.8, 0.9]$. The above changes are not part of this thesis and are described exclusively (with the solution of the LP) inside the paper [2].

First conclusions about Fairness in NFRS

This post-processing analysis aims to answer important questions about fairness in NF-RS. They inspect the trade-off between the gains that can be achieved by a NF-RS algorithm and how this will affect the fairness of the system. Moreover, they continue to try to design a network-friendly algorithm that aims to deliver maximum network gain under a fairness constraint. The design of such an algorithm requires a mathematical approach to the problem, which is extensively explained and modeled as an LP problem.

The plots and the data generated by this work have shown many important details about the significance of the Fair-NFRS system which can be proven that may perform very well in relation with other NFR algorithms and can also achieve network gains by allowing some little unfairness. Most of the original assumptions have been verified but this thesis will propose some more metrics to help in this verification.

3.4 Content Bubble and Fairness

While the previous research results are promising regarding fairness in a network-friendly recommendation system, there is a significant concern about the presence of diversity in the new recommendations. We will initially analyze and comprehend the nature of the diversity issue, referred to as the **Content bubble**.

Inspecting diversity as another aspect of Fairness in NF-RS

Diversity guarantees that users experience a broad selection of items that align with their preferences and interests. Diverse recommendations introduce users to new and potentially relevant content, enhancing their overall experience by avoiding repeatedly suggesting similar items.

The recommendation system has a very clear mechanism for how it selects its suggestions. As mentioned before RS use recommendation lists related to items that are based on probabilities. In our scenario, the baseline RS includes recommendations that follow a top caching policy (4.7) which means that cached contents are prioritized. The recommendation list includes probabilities p_i for content i without considering it is direct from the user or comes from the recommendation system.
In order to examine the diversity of the recommendations we will need to analyze the distribution of the probabilities inside the probability demand vectors. Distribution in RS refers to how items are spread or allocated across the recommendation space. It represents the relative probabilities of different items being recommended to users. The distribution and variability of these probability scores illustrate the broad range and uncertainty in the recommendations.

So by considering the above analysis, we would search for metrics that capture the "uncertainty" inside a distribution, as it was explained before, and try to quantify the level of diversity provided by the recommendation system.

After presenting the above metrics, we will now apply them to the data that we were given by the researchers in order to evaluate diversity in Fair-NFRS. In the next Chapter, we will analyze our results.

Chapter 4

Interpreting Results

4.1 Expectations and assumptions

Motivation

The related work of [2] that we described in the previous chapter, has demonstrated that it is possible to still achieve good cost reductions (as in the original 'non-fair' NF-RS algorithms), with only small levels of intrusiveness or unfairness, as measured by their proposed metrics. The unfairness between the recommended items of Baseline and the network-friendly recommender is captured by the metrics in the paper [2]. In this set of experiments, our goal is to measure how both the baseline as well as the NF-RS algorithms affect our own proposed metrics that capture *Content bubble* phenomena. By utilizing Entropy as a metric of unpredictability, we will examine how diversity as another fairness aspect is affected by the NF-RS algorithms. The Gini impurity index serves a similar purpose since it quantifies the extent of dispersion and imbalance in the probability distribution when considering network profits under fairness limitations.

4.2 Data set

As stated in Sections 2, 3, the data used for the experiments regarding fairness are drawn from two datasets that were briefly described: ***LastFm*** and ***MovieLens***.

A subset of the many scenarios evaluated in the paper will be analyzed in this thesis. The paper *Fairness in Network Friendly Recommendations* is utilizing a variety of metrics to measure the quantity of fairness or unfairness on recommendations that are network-friendly.
After forming the U matrices from the two datasets the researchers applied network friendly algorithms in order to produce new recommendation lists. Two categories of recommendation lists are being compared and are the basis of this

thesis. Baseline and Network friendly recommendations are the key components of this research and every algorithm or metric is focused on them.

Datasets

The paper extracts its data from two main datasets: LastFm and MovieLens.

1. For the first one , **LastFm**, they applied *getSimilar* method to the content ID's and fill the matrix U containing scores u_{ij}. The matrix U is quite sparse and to deal with this situation they retained the most significant component of the underlying graph and set values above a threshold of 0.1 in the matrix U to 1.

2. For the **MovieLens** dataset they utilized an item-to-item Collaborative filtering method with the 10 most similar items and extracted the missing users' ratings. The similarity between content pairs was determined using *coisine distance* and for pairs with a distance greater than 0.6 u_{ij} is set 1. If the distance is less than 0.6 then the u_{ij} is set to 0.

The datasets that were described above will now be used by some NFRS algorithms in order to generate results for the paper's research about Fairness.

4.2.1 Multi-Step Algorithm

Multi-step NF-RS [29] is an algorithm that returns the optimal solution in our model setup under no fairness requirements. It does this by including in each recommendation list R_i a set of contents that satisfy a quality of service constraint, which is similar to the Greedy NF-RS. Additionally, it maximizes the network gains over the long term by taking into account requests made directly and through recommendations, as well as the probability a.

After applying the above network-friendly algorithm to these sets of data, some files are generated containing info about the new network-friendly recommendation lists R^{NF}. These probability lists are part of archives that were produced by processing several scenarios that each represent a very particular network state.

Optimal Fair-NFRS

After applying the Multi-Step algorithm, the FAIR-NFR algorithm will assign fairness constraints in order to optimize the recommendations considering also fairness. The form of the archives is presented in 4.7 where the Fairness Constraints are also presented in 4.1.

4.3 Metrics for evaluating the impact of fairness on diversity

This thesis aims to enhance the research conducted in the previous foundational paper by proposing three novel measures for assessing the fairness of network-friendly recommendations, as outlined in [2]. This paper is the first to examine fairness in network-friendly recommendation systems and develop a FAIR-NFRS. Based on the assumptions of section 3.4 we will propose the following metrics in order to capture diversity and to address the existence of Content Bubble in FAIR-NFRS.

The three new metrics that this thesis suggests are :

1. **Entropy**
2. **Gini Impurity Index**
3. **Variance**

4.4 Metric 1: Entropy

One of the most important definitions of entropy inside the world of statistics is that entropy, according to *shannon*, is a measure of information (or *measure of uncertainty*) and this is the definition we are going to obey. We will apply entropy in order to measure the fairness and diversity in the recommendation lists that are promoted by the network-friendly system. Based on the 3 fairness functions that were discussed on Chapter 2 and the results that came up after being applied to different scenarios, Entropy will be the metric that will help us measure the (un)fairness created by the NFR algorithms and also verify the original assumptions about fairness in this kind of systems.

What is Entropy

The term *Entropy* can have varying interpretations depending on the specific context in which it is employed. Entropy is commonly utilized and regarded as a metric for randomness, uncertainty, or disorder in various domains, including thermodynamics, statistics, and machine learning [40].

Entropy in Information Theory

Entropy in information theory has a special role in measuring information. Information could be described in terms of the probabilities of events, meaning a probability distribution.

Let X be a discrete random variable.

The probability mass function (PMF) of a discrete random variable X is often denoted as $P(X = x)$. For a given set of possible values $x_1, x_2, \ldots, x_n$, the PMF is defined as [41]:

$$P(X = x_i) = \text{probability of } x_i \text{ for } i = 1, 2, \ldots, n$$

- $H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$.

4.4.1 Algorithm of entropy

In our case we will use probability vectors that contain the information about $\mathbf{p} = [p_1, \ldots, p_K]$ where p is the vector with the distribution of total demand for all contents.

In this situation, we implement the formula of entropy as :

- $Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$.

Algorithm 1: Calculate Entropy

Data: Probability distribution P
Result: Entropy $Entropy$

$Entropy \leftarrow 0$;
for $i \leftarrow 1$ **to** n **do**
  $Entropy \leftarrow Entropy - p_i \log_2(p_i)$;
end
return $Entropy$;

4.5 Metric 2: Gini Impurity Index

Gini is a metric that utilizes the logic and algorithm of Entropy but in a different way. It is a metric that is commonly used in classification problems due to its ability to analyze feature selection [42].

- *Gini Index Impurity is a metric to measure the frequency of a random item to be misidentified.*

Measuring inequality of a distribution, Gini Impurity index has a range of values $[0, 1]$ where:

- 0: indicating absolute purity
- 1: indicating absolute impurity

Gini Impurity Index Formula is :

- **Gini** $= 1 - \sum_{i=1}^{n} p_i^2$.

1. p_i is the probability of demand for content i
2. n is is the total number of contents.

What is Gini Impurity Index

In decision trees or machine learning, Gini Index is a metric used for measuring impurity or inequality of a distribution. Low values of Gini index indicate a more homogenous distribution and high values of Gini the opposite. In classification problems and especially in decision trees, Gini, is used for examining the quality of the split by analyzing the differences between the inequality of parent and child nodes. In splitting, the algorithms aim to split the data in a way that a low score in Gini is achieved at each node. A low Gini index indicates a high-quality split.

Gini IMID in RS

In recommendation systems, Gini impurity is a metric commonly employed to evaluate the variety of recommendations. In the context of a recommendation system, Gini impurity may be employed as a metric to assess the degree of imbalance in the distribution of recommended items. The presence of a lower Gini impurity might indicate an even distribution of recommendations.

Gini impurity can be employed to quantify the uncertainty or impurity of a distribution in probability vectors. A lower Gini impurity would suggest a probability distribution that is more certain or pure.

4.5.1 Algorithm of Gini Impurity Index

Algorithm 2: Calculate Gini Impurity Index

Data: Probability distribution P
Result: Gini impurity index G

$G \leftarrow 1$;
for $i \leftarrow 1$ **to** n **do**
  $p_i \leftarrow$ probability demand of content i ;
  $G \leftarrow G - p_i^2$;
end
return G;

4.6 Metric 3: Variance as a measure for diversity

One of our initial options for measuring diversity in recommendations was ***Variance***. This metric is commonly utilized for measuring the spread of probabilities inside a probability vector. Based on this assumption, we thought that by using Variance on the NF-RS probability vector, we could evaluate diversity. The variance is the average of the squared deviations from the mean. It indicates how much the values in a dataset deviate from the mean.

- $\text{Var} = \frac{1}{n-1}\sum_{i=1}^{n}(p_i - \bar{p})^2$
  where n is the length of the vector p.

Although equipping variance to measure diversity seemed to be a reasonable strategy, the outcomes were perplexing as we elaborate in the next Chapter.

4.6.1 Entropy on fairness

The files with fairness constraints have the same structure as the ones we previously discussed. We will use the vectors with the distribution of total demand for all contents for both Baseline P^{BS} and network-friendly recommendations P^{NF}.

So the formula of entropy will be as follows :

- Baseline Entropy (Entropy_bs): $Entropy\_BS = -\sum_{i=1}^{n} p_i^{BS} \cdot \log_2 p_i^{BS}$.
- Network Friendly RS Entropy (Entropy_NF):

  $Entropy\_NF = -\sum_{i=1}^{n} p_i^{NF} \cdot \log_2 p_i^{NF}$.

For smoother results both entropies will be normalized by dividing them with the logarithmic base of the length of every vector:

- $Entropy\_NF$ divided by $----> :$ $log(len(P^{NF}))$
- $Entropy\_BS$ divided by $----> :$ $log(len(P^{BS}))$

4.6.2 Gini Impurity Index on fairness

The same changes will be made for the formula of Gini Impurity Index .

- Baseline Gini Impurity Index (Gini_bs) $Gini\_BS = 1 - \sum_{i=1}^{n} {p_i^{BS}}^2$.
- Network Friendly Gini Impurity Index (Gini_nf) $Gini\_NF = 1 - \sum_{i=1}^{n} {p_i^{NF}}^2$.

Measuring Content Bubble

After applying the metrics that we have previously presented, Entropy and Gini Impurity Index (we will explain later why variance is missing), we will now examine and observe their results. This thesis has a very specific purpose and that is to determine how fairness could affect the diversity of the recommendations, potentially leading to a lack of novelty or serendipity for users. Every suggested item is linked to a probability score that indicates the user's likelihood of finding it relevant or preferable. The spread and variability of these probability scores demonstrate the wide range and lack of certainty in the suggestions.

We will basically analyze how entropy is affected by fairness constraints and what this tells us about the diversity of recommendations. The entropy value quantifies the level of uncertainty or randomness in the distribution of recommendations. Greater entropy levels indicate greater diversity and equilibrium, whereas lower entropy values imply lower diversity and potentially biased recommendations.

The Gini impurity index is computed by considering the probability distribution of recommended items. The probability of each item contributes to the overall impurity of the recommendation set. A higher Gini impurity index signifies increased diversity or imbalance in the probability distribution, indicating a more diverse set of recommendations where probabilities are distributed throughout a broader variety of items.

4.7 Scenarios

Every scenario is described by the following attributes :

- $\boldsymbol{U}$: which are the matrices extracted from the datasets of LastFm and MovieLens as they were analyzed in previous sections.
- Popularity (***pop***): Items are classified as popular (assigned a value of one) or non-popular (assigned a value of zero) in a binary popularity-based caching policy. The caching system then gives priority to storing and retrieving items with a value of one, with the goal of maximizing performance by concentrating on content that is frequently accessed while ignoring less popular items.
- $\boldsymbol{a}$:which is the probability a user follow the recommended item $a \in [0.5, 0.8, 0.99]$
- $\boldsymbol{N}$: describes the total number of recommendations $[2, 5]$
- $\boldsymbol{C}$: Cache size values $C \in [5, 10, 20]$
- ***CP***: caching policy and we only use *top* cp in the 'top' policy the C contents with the highest pBS demand are cached

- ***QQ***: Quality of recommendations constraint $q \in [0.5, 0.8, 0.9]$
- ***L***: L=40 or L=1 . We use L=40 for the sequential NF optimal algorithm (Multi-step algorithm)

- ***F_*** : Fairness Constraints

The new scenarios created include the fairness functions f_{max},f_{tv}, and f_{kl} and the fairness constraints that were assigned to them.

The constraints that will be used in this thesis are the following:

F_{MAX}	$[0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30]$
F_{TV}	$[0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.60, 0.80]$
F_{KL}	$[0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.45, 0.6]$

Table 4.1: Fairness Constraints

4.8 Applying the new Metrics

We will now utilize the generated files with fairness constraints in order to verify the paper's assumptions about fairness. *Entropy* and *Gini impurity index* will provide results about how the network-friendly profile of a recommendation system that ensures fairness affects the diversity of recommendations in comparison with the baseline RS.

Baseline Comparisons

The results of the new metrics will be compared with the results of the baseline recommendation system which will represent the best possible outcomes. Both *LastFm* and *MovieLens* datasets provided data with u_{ij} scores (exhibiting a recommendation score on how good recommendation j is after consuming content i). These scores were generated from a simple Collaborative filtering model as it is described in the paper [2].

As presented in the subsection (4.7), the new archives that describe a specific network state (U,pop,a,N,C,CP,Q,L) will now include the fairness function that was used on them and the fairness constraint that the table (4.1) includes.

e.g $Ulastfm\_pop0\_a0.99\_N2\_C5\_CPtop\_Q0.9\_L40\_FKL0.01$

4.9 Results and Figures

1. **LastFM dataset with parameters pop=0,a=0.99, N=2,C=20,q=0.5,L=40**

entropy_BS	gini_BS
0.97281918	0.998044

Figure 4.1: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
lastfm	0	0.8	5	20	top	0.5	40	0.941458085	0.996325	0.113955852	KL	0.01
lastfm	0	0.8	5	20	top	0.5	40	0.864887479	0.990136	0.302087472	KL	0.05
lastfm	0	0.8	5	20	top	0.5	40	0.785228788	0.981326	0.452170205	KL	0.1
lastfm	0	0.8	5	20	top	0.5	40	0.702772424	0.960475	0.55127564	KL	0.15
lastfm	0	0.8	5	20	top	0.5	40	0.634384852	0.93803	0.598041997	KL	0.2
lastfm	0	0.8	5	20	top	0.5	40	0.623831122	0.936255	0.601771501	KL	0.25
lastfm	0	0.8	5	20	top	0.5	40	0.623831122	0.936255	0.601771501	KL	0.3
lastfm	0	0.8	5	20	top	0.5	40	0.917863939	0.994711	0.2	max	0.01
lastfm	0	0.8	5	20	top	0.5	40	0.673344587	0.964982	0.571071412	max	0.05
lastfm	0	0.8	5	20	top	0.5	40	0.6287154	0.941388	0.598321449	max	0.1
lastfm	0	0.8	5	20	top	0.5	40	0.623831122	0.936255	0.601771501	max	0.15
lastfm	0	0.8	5	20	top	0.5	40	0.623831122	0.936255	0.601771501	max	0.2
lastfm	0	0.8	5	20	top	0.5	40	0.623831122	0.936255	0.601771501	max	0.25
lastfm	0	0.8	5	20	top	0.5	40	0.623831122	0.936255	0.601771501	max	0.3
lastfm	0	0.8	5	20	top	0.5	40	0.971650613	0.998	0.01	TV	0.01
lastfm	0	0.8	5	20	top	0.5	40	0.965521298	0.997659	0.05	TV	0.05
lastfm	0	0.8	5	20	top	0.5	40	0.951288952	0.996643	0.1	TV	0.1
lastfm	0	0.8	5	20	top	0.5	40	0.931813165	0.994826	0.15	TV	0.15
lastfm	0	0.8	5	20	top	0.5	40	0.909932768	0.992584	0.2	TV	0.2
lastfm	0	0.8	5	20	top	0.5	40	0.883103914	0.989079	0.25	TV	0.25
lastfm	0	0.8	5	20	top	0.5	40	0.853378751	0.984724	0.3	TV	0.3

Figure 4.2: Arithmetic Results

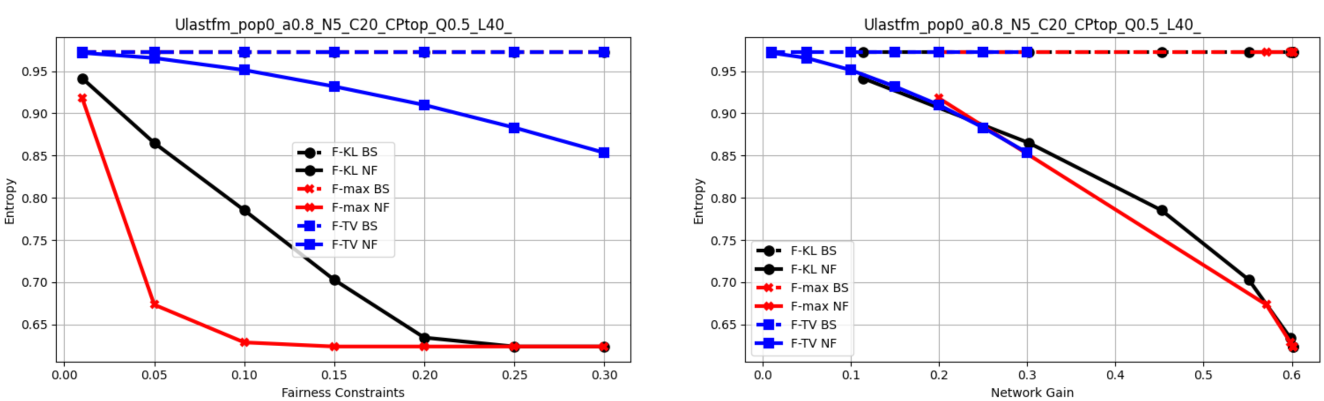

Figure 4.3: Entropy in relation with Fairness Constraints and Network Gain

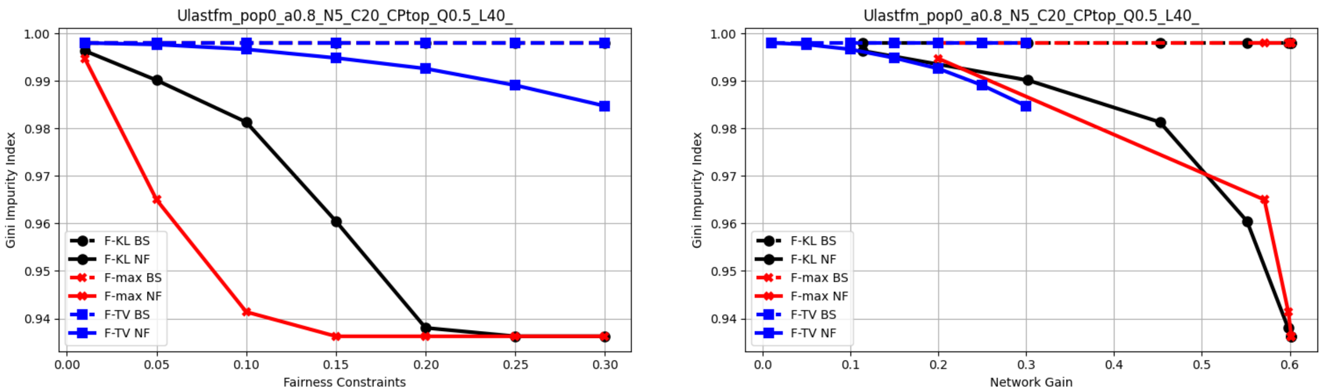

Figure 4.4: Gini Index in relation with Fairness Constraints and Network Gain

Observations 1

As the fairness constraints ascend , Entropy of the system does not seem able to keep up and begins a descending behavior .

- Range of constraints $\in [0, 0.30]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results and indicate that for low values of fairness constraints the Fair-NFRS could reach the behavior of Baseline .
- For the values that the two metrics align with the baseline , the system seems to achieve the lowest network gains

2. **LastFM dataset with parameters pop=0,a=0.99, N=2,C=5,q=0.9,L=40**

In this scenario we decrease the Cache size and increase q constraint :
C(20 $\rightarrow$ 5) , q(0.5 $\rightarrow$ 0.9)

Reducing the Cache size limits the system's ability to store more content and increasing the q constraint could result into more personalized recommendations but with less diversity.

entropy_BS	gini_BS
0.876118	0.995781

Figure 4.5: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
lastfm	0	0.99	2	5	top	0.9	40	0.826773745	0.990436	0.123498521	KL	0.01
lastfm	0	0.99	2	5	top	0.9	40	0.745791276	0.976344	0.253404472	KL	0.05
lastfm	0	0.99	2	5	top	0.9	40	0.601424589	0.937263	0.389537886	KL	0.15
lastfm	0	0.99	2	5	top	0.9	40	0.424351385	0.84697	0.484674268	KL	0.3
lastfm	0	0.99	2	5	top	0.9	40	0.338004353	0.785781	0.510258782	KL	0.4
lastfm	0	0.99	2	5	top	0.9	40	0.316707765	0.773579	0.516139109	KL	0.45
lastfm	0	0.99	2	5	top	0.9	40	0.307346288	0.770495	0.518490976	KL	0.6
lastfm	0	0.99	2	5	top	0.9	40	0.916692407	0.995283	0.05	max	0.01
lastfm	0	0.99	2	5	top	0.9	40	0.74143569	0.97461	0.25	max	0.05
lastfm	0	0.99	2	5	top	0.9	40	0.446243517	0.92057	0.464359108	max	0.1
lastfm	0	0.99	2	5	top	0.9	40	0.39432025	0.888121	0.509167105	max	0.15
lastfm	0	0.99	2	5	top	0.9	40	0.381924348	0.866735	0.513695038	max	0.2
lastfm	0	0.99	2	5	top	0.9	40	0.333628529	0.800088	0.51729483	max	0.25
lastfm	0	0.99	2	5	top	0.9	40	0.872431338	0.995517	0.01	TV	0.01
lastfm	0	0.99	2	5	top	0.9	40	0.84233419	0.991701	0.1	TV	0.1
lastfm	0	0.99	2	5	top	0.9	40	0.793741774	0.983718	0.197399875	TV	0.2
lastfm	0	0.99	2	5	top	0.9	40	0.670757627	0.960069	0.323821742	TV	0.4
lastfm	0	0.99	2	5	top	0.9	40	0.525967773	0.90933	0.439221899	TV	0.6
lastfm	0	0.99	2	5	top	0.9	40	0.351694795	0.796009	0.509455706	TV	0.8

Figure 4.6: Arithmetic Results

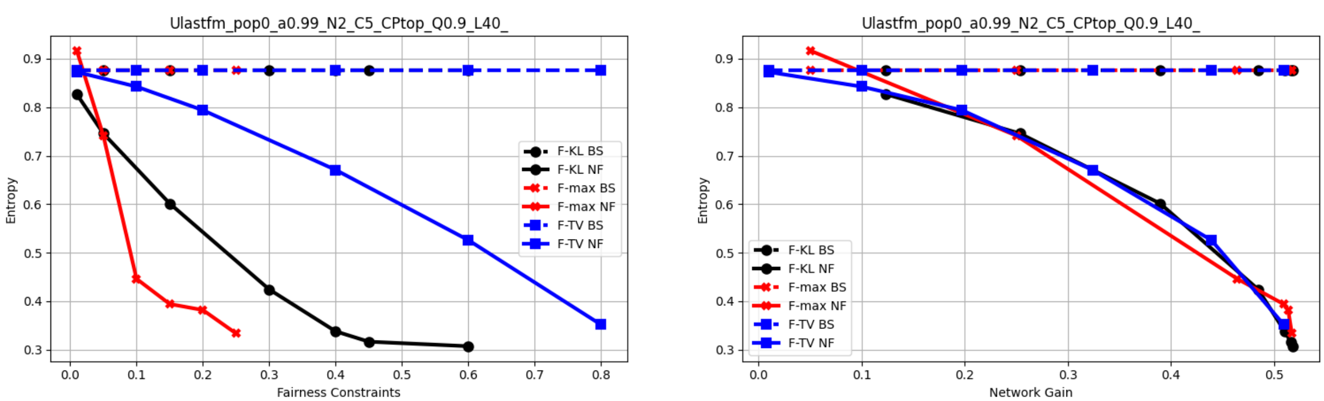

Figure 4.7: Entropy in relation with Fairness Constraints and Network Gain

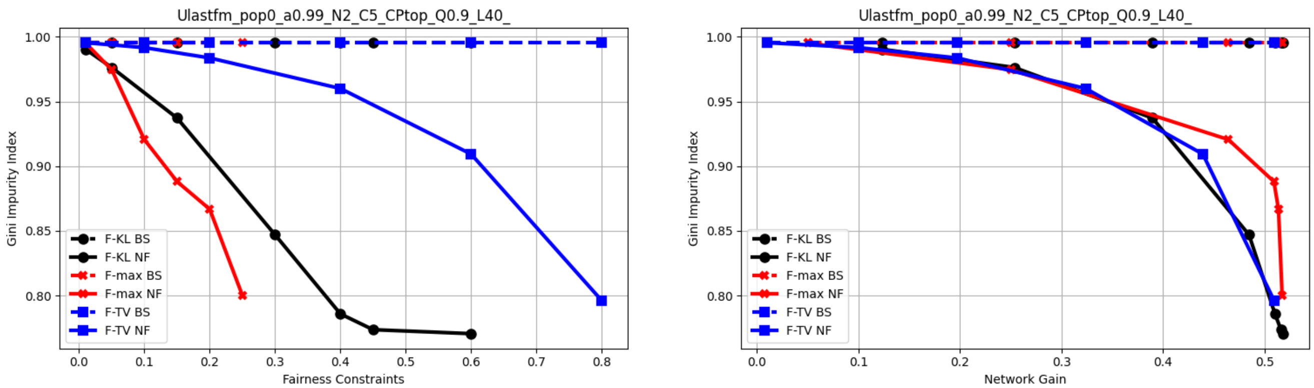

Figure 4.8: Gini Index in relation with Fairness Constraints and Network Gain

Observations 2

As the fairness constraints ascend , Entropy of the system does not seem able to keep up and begins a descending behavior .

- Range of constraints $\in [0, 0.8]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results and indicate that for low values of fairness constraints the Fair-NFRS could reach the behavior of Baseline . Although the range of values is wider, for higher fairness constraints the scores of the metrics drop dramatically .
- For the values that the two metrics align with the baseline , the system seems to achieve the lowest network gains.

3. **LastFM dataset with parameters pop=0,a=0.99, N=2,C=10,q=0.8,L=40**

In this scenario we increase the Cache size and decrease q constraint :
C(5 $\rightarrow$ 10) , q(0.9 $\rightarrow$ 0.8)

By increasing cache size and reducing the quality constraint, there are no significant changes compared to the previous scenario (for this smaller range of fairness constraints).

entropy_BS	gini_BS
0.876118	0.995781

Figure 4.9: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
lastfm	0	0.99	2	10	top	0.8	40	0.824618688	0.991275	0.144524629	KL	0.01
lastfm	0	0.99	2	10	top	0.8	40	0.742739712	0.978406	0.322966333	KL	0.05
lastfm	0	0.99	2	10	top	0.8	40	0.656265897	0.956949	0.463337086	KL	0.1
lastfm	0	0.99	2	10	top	0.8	40	0.582176929	0.934004	0.557302857	KL	0.15
lastfm	0	0.99	2	10	top	0.8	40	0.52187359	0.910946	0.621214791	KL	0.2
lastfm	0	0.99	2	10	top	0.8	40	0.469604715	0.886322	0.667660939	KL	0.25
lastfm	0	0.99	2	10	top	0.8	40	0.423628629	0.861759	0.701978955	KL	0.3
lastfm	0	0.99	2	10	top	0.8	40	0.883933794	0.993696	0.1	max	0.01
lastfm	0	0.99	2	10	top	0.8	40	0.583021795	0.958414	0.5	max	0.05
lastfm	0	0.99	2	10	top	0.8	40	0.42403189	0.915817	0.690632327	max	0.1
lastfm	0	0.99	2	10	top	0.8	40	0.395191617	0.893437	0.717421267	max	0.15
lastfm	0	0.99	2	10	top	0.8	40	0.365338287	0.856742	0.737184097	max	0.2
lastfm	0	0.99	2	10	top	0.8	40	0.314033627	0.794175	0.753051784	max	0.25
lastfm	0	0.99	2	10	top	0.8	40	0.268980721	0.752042	0.759072049	max	0.3
lastfm	0	0.99	2	10	top	0.8	40	0.872049241	0.995561	0.01	TV	0.01
lastfm	0	0.99	2	10	top	0.8	40	0.865099423	0.994888	0.05	TV	0.05
lastfm	0	0.99	2	10	top	0.8	40	0.849878962	0.993148	0.1	TV	0.1
lastfm	0	0.99	2	10	top	0.8	40	0.835026789	0.989989	0.15	TV	0.15
lastfm	0	0.99	2	10	top	0.8	40	0.783620853	0.983855	0.2	TV	0.2
lastfm	0	0.99	2	10	top	0.8	40	0.759389631	0.977758	0.25	TV	0.25
lastfm	0	0.99	2	10	top	0.8	40	0.725981338	0.971248	0.3	TV	0.3

Figure 4.10: Arithmetic Results

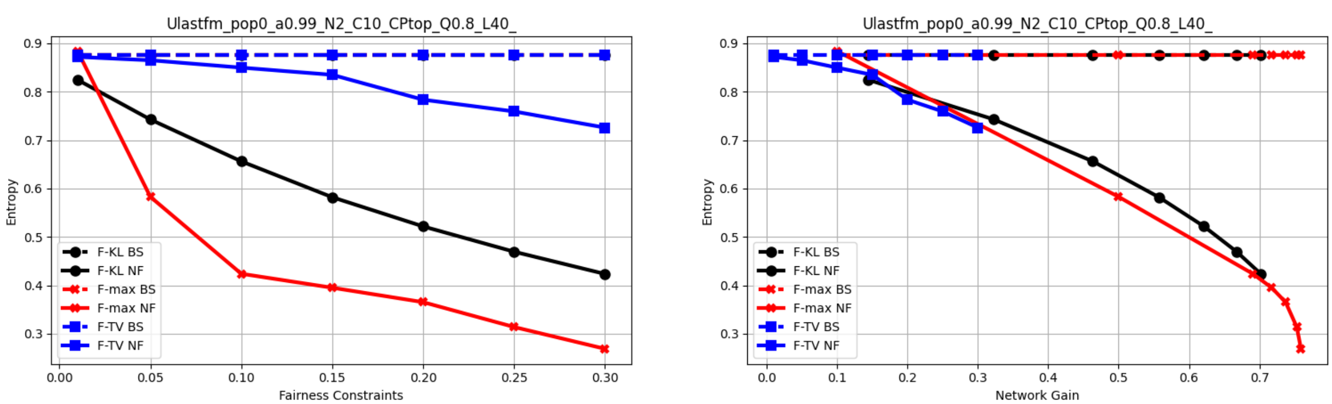

Figure 4.11: Entropy in relation with Fairness Constraints and Network Gain

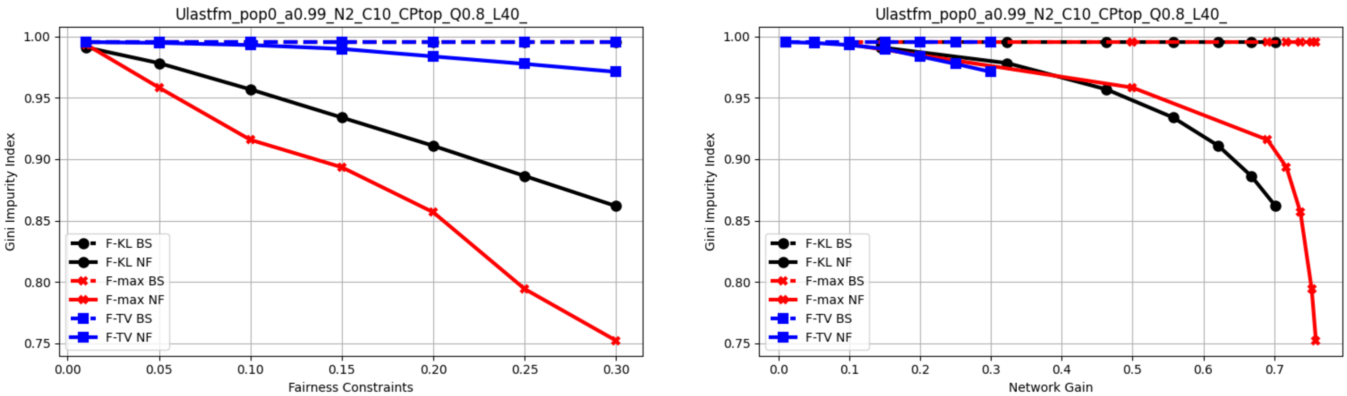

Figure 4.12: Gini Index in relation with Fairness Constraints and Network Gain

Observations 3

- Range of constraints $\in [0, 0.30]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results and indicate that for low values of fairness constraints, the Fair-NFRS could reach the behavior of Baseline .
- For the values that the two metrics align with the baseline , the system seems to achieve the lowest network gains

4. **LastFM dataset with parameters pop=0,a=0.99, N=2,C=10,q=0.9,L=40**

In this scenario we increase the value of the quality constraint :
q(0.8 $\rightarrow$ 0.9)

By increasing only the q constraint we should see results that are on the same level of Entropy as the previous scenario. For the same range of fairness constraints this scenario performs the same as the previous scenario although as the value of the constraint increases, entropy drops dramatically.

entropy_BS	gini_BS
0.876118	0.995781

Figure 4.13: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
lastfm	0	0.99	2	10	top	0.9	40	0.824761503	0.991193	0.143438442	KL	0.01
lastfm	0	0.99	2	10	top	0.9	40	0.74462434	0.979537	0.305508004	KL	0.05
lastfm	0	0.99	2	10	top	0.9	40	0.606466916	0.95077	0.472262273	KL	0.15
lastfm	0	0.99	2	10	top	0.9	40	0.451678563	0.890241	0.578044537	KL	0.3
lastfm	0	0.99	2	10	top	0.9	40	0.368949193	0.851664	0.608176301	KL	0.45
lastfm	0	0.99	2	10	top	0.9	40	0.366275384	0.850532	0.608502483	KL	0.6
lastfm	0	0.99	2	10	top	0.9	40	0.888718993	0.993816	0.1	max	0.01
lastfm	0	0.99	2	10	top	0.9	40	0.522476479	0.954232	0.495640856	max	0.05
lastfm	0	0.99	2	10	top	0.9	40	0.468938759	0.926009	0.566246134	max	0.1
lastfm	0	0.99	2	10	top	0.9	40	0.409857122	0.884964	0.588461202	max	0.15
lastfm	0	0.99	2	10	top	0.9	40	0.374298519	0.857136	0.606308171	max	0.2
lastfm	0	0.99	2	10	top	0.9	40	0.366264672	0.850534	0.608512771	max	0.25
lastfm	0	0.99	2	10	top	0.9	40	0.871469566	0.9955	0.01	TV	0.01
lastfm	0	0.99	2	10	top	0.9	40	0.849898189	0.992449	0.1	TV	0.1
lastfm	0	0.99	2	10	top	0.9	40	0.79011166	0.985411	0.2	TV	0.2
lastfm	0	0.99	2	10	top	0.9	40	0.679228294	0.970104	0.387775022	TV	0.4
lastfm	0	0.99	2	10	top	0.9	40	0.544026187	0.935972	0.527324619	TV	0.6
lastfm	0	0.99	2	10	top	0.9	40	0.396825925	0.866983	0.603849713	TV	0.8

Figure 4.14: Arithmetic Results

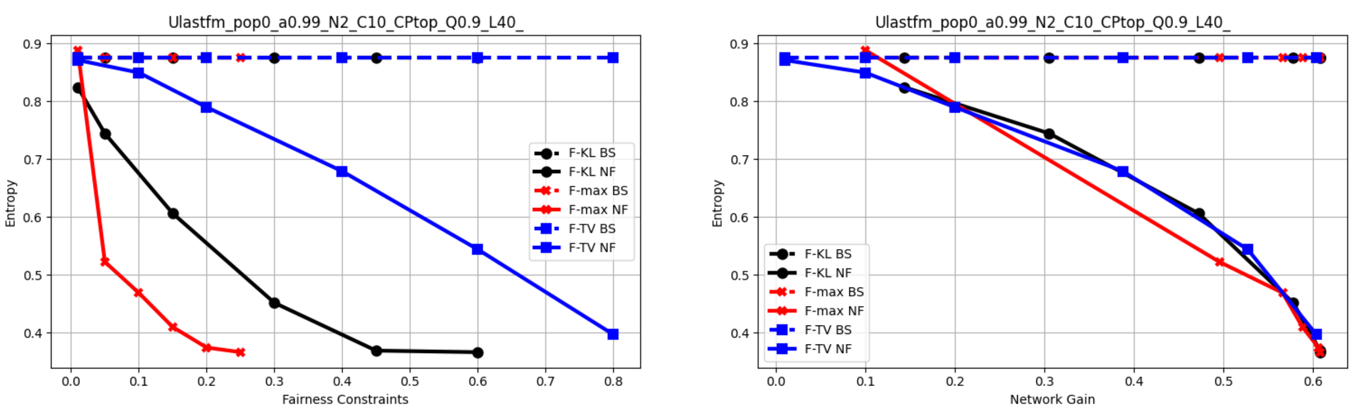

Figure 4.15: Entropy in relation with Fairness Constraints and Network Gain

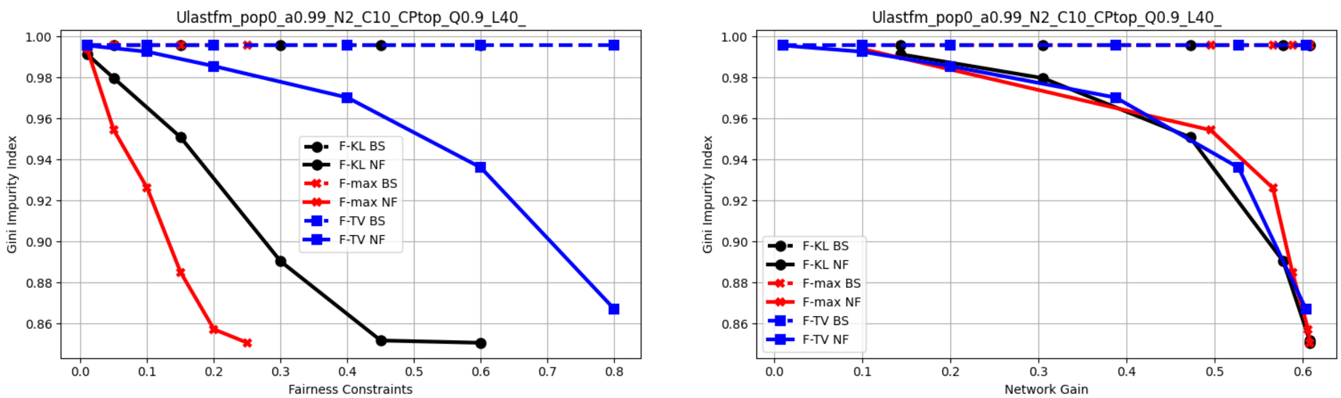

Figure 4.16: Gini Index in relation with Fairness Constraints and Network Gain

Observations 4

- Range of constraints $\in [0, 0.80]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- The system appears to achieve the least amount of network gain for the values where both of the metrics align with the baseline results.
- Network Gains $\uparrow - - -- > Gini, Entropy \downarrow$

5. **LastFM dataset with parameters pop=0,a=0.99, N=5,C=10,q=0.8,L=40**

In this scenario we increase the size of the recommendation list and decrease the q constraint :
N(2 $\rightarrow$ 5) , q(0.9 $\rightarrow$ 0.8)

A larger recommendation list indicates more suggestions.The q constraint decreases, but it is still high. These changes lead to an increase in the behavior of entropy compared to the range of constraints of the previous scenario.

entropy_BS	gini_BS
0.925228	0.996883

Figure 4.17: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
lastfm	0	0.99	5	10	top	0.8	40	0.868959071	0.993211	0.133183241	KL	0.01
lastfm	0	0.99	5	10	top	0.8	40	0.774192619	0.980187	0.302312748	KL	0.05
lastfm	0	0.99	5	10	top	0.8	40	0.678276519	0.954942	0.405694037	KL	0.1
lastfm	0	0.99	5	10	top	0.8	40	0.609754923	0.941234	0.468286282	KL	0.15
lastfm	0	0.99	5	10	top	0.8	40	0.557922381	0.930973	0.508034438	KL	0.2
lastfm	0	0.99	5	10	top	0.8	40	0.518828529	0.923269	0.531889211	KL	0.25
lastfm	0	0.99	5	10	top	0.8	40	0.487619721	0.917298	0.545852777	KL	0.3
lastfm	0	0.99	5	10	top	0.8	40	0.923448091	0.99501	0.1	max	0.01
lastfm	0	0.99	5	10	top	0.8	40	0.578689663	0.962523	0.4331503	max	0.05
lastfm	0	0.99	5	10	top	0.8	40	0.485817377	0.928877	0.527876416	max	0.1
lastfm	0	0.99	5	10	top	0.8	40	0.461178166	0.91349	0.553480525	max	0.15
lastfm	0	0.99	5	10	top	0.8	40	0.461178644	0.91349	0.553480213	max	0.2
lastfm	0	0.99	5	10	top	0.8	40	0.46117854	0.91349	0.553480283	max	0.25
lastfm	0	0.99	5	10	top	0.8	40	0.461178644	0.91349	0.553480213	max	0.3
lastfm	0	0.99	5	10	top	0.8	40	0.922835982	0.996746	0.01	TV	0.01
lastfm	0	0.99	5	10	top	0.8	40	0.916483537	0.996038	0.05	TV	0.05
lastfm	0	0.99	5	10	top	0.8	40	0.900943277	0.994042	0.1	TV	0.1
lastfm	0	0.99	5	10	top	0.8	40	0.879621113	0.991041	0.15	TV	0.15
lastfm	0	0.99	5	10	top	0.8	40	0.853234474	0.986504	0.2	TV	0.2
lastfm	0	0.99	5	10	top	0.8	40	0.82268691	0.981271	0.25	TV	0.25
lastfm	0	0.99	5	10	top	0.8	40	0.777963436	0.975804	0.3	TV	0.3

Figure 4.18: Arithmetic Results

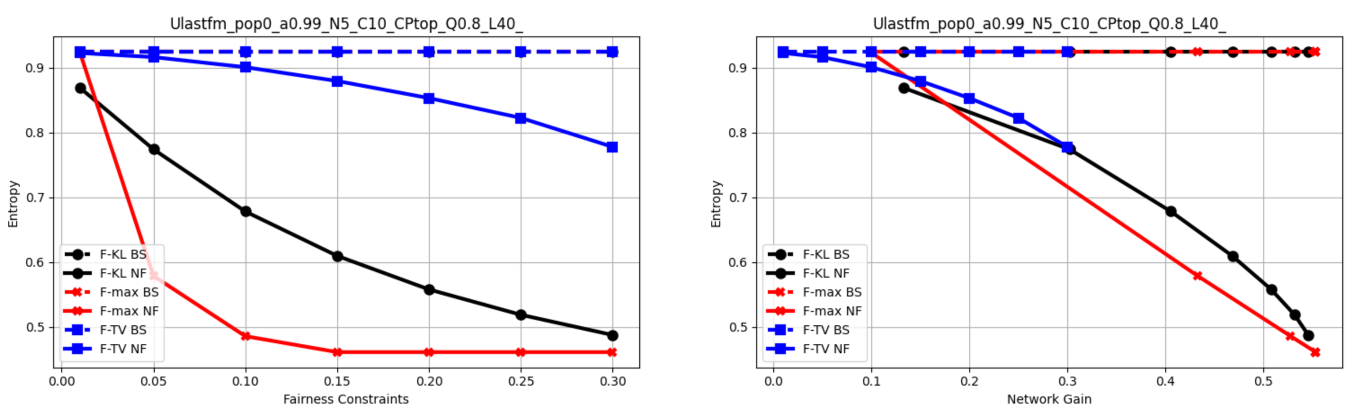

Figure 4.19: Entropy in relation with Fairness Constraints and Network Gain

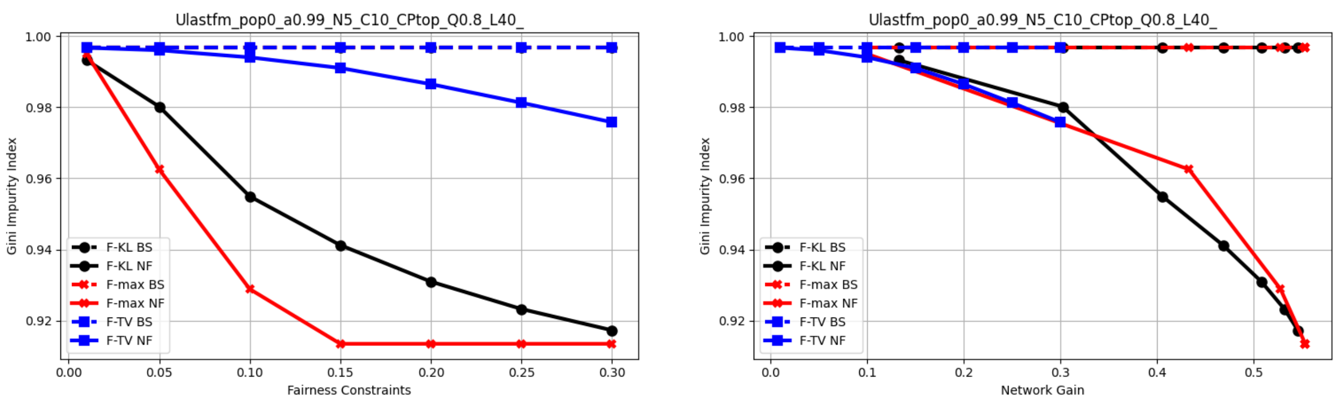

Figure 4.20: Gini Index in relation with Fairness Constraints and Network Gain

Observations 5

- Range of constraints $\in [0, 0.3]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results.
  Fairness Constraints $\uparrow - - -- > Gini, Entropy \downarrow$
- Again here higher network gains are overshadowed by very low metric values.
  Network Gains $\uparrow - - -- > Gini, Entropy \downarrow$

6. **LastFM dataset with parameters pop=1,a=0.8, N=2,C=10,q=0.5,L=40**

In this scenario, we increase the size of the recommendation list, decrease the q constraint,assign value 1 to popularity, and decrease the probability (a) of the user following the recommendation:
N(5 $\rightarrow$ 2) , q(0.8 $\rightarrow$ 0.5),pop(0 $\rightarrow$ 1) , a(0.99 $\rightarrow$ 0.8)

By assigning the value of 1 to popularity, this makes the items popular. The size of the recommendation list shrinks, and the quality constraint decreases. The probability of the user following the recommended item also decreases. By having only 2 items to recommend, and in combination with the popularity bias, the recommendations will lack variety. As shown in the plots, for the given fairness constraints, the metrics achieve lower values compared to the previous scenario.

entropy_BS	gini_BS
0.892692	0.995259

Figure 4.21: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
lastfm	1	0.8	2	10	top	0.5	40	0.830736784	0.988469	0.149038188	KL	0.01
lastfm	1	0.8	2	10	top	0.5	40	0.738950632	0.972706	0.330318527	KL	0.05
lastfm	1	0.8	2	10	top	0.5	40	0.652151232	0.95302	0.46819214	KL	0.1
lastfm	1	0.8	2	10	top	0.5	40	0.581395139	0.930283	0.55876628	KL	0.15
lastfm	1	0.8	2	10	top	0.5	40	0.526202846	0.906568	0.613727648	KL	0.2
lastfm	1	0.8	2	10	top	0.5	40	0.49548946	0.88782	0.632740921	KL	0.25
lastfm	1	0.8	2	10	top	0.5	40	0.47652482	0.874932	0.634880923	KL	0.3
lastfm	1	0.8	2	10	top	0.5	40	0.897158138	0.992582	0.1	max	0.01
lastfm	1	0.8	2	10	top	0.5	40	0.614029524	0.956381	0.5	max	0.05
lastfm	1	0.8	2	10	top	0.5	40	0.518369585	0.915898	0.629733071	max	0.1
lastfm	1	0.8	2	10	top	0.5	40	0.479738672	0.881615	0.634880043	max	0.15
lastfm	1	0.8	2	10	top	0.5	40	0.473664667	0.869305	0.634880043	max	0.2
lastfm	1	0.8	2	10	top	0.5	40	0.459600566	0.849591	0.634880043	max	0.25
lastfm	1	0.8	2	10	top	0.5	40	0.459600566	0.849591	0.634880043	max	0.3
lastfm	1	0.8	2	10	top	0.5	40	0.888523212	0.995011	0.01	TV	0.01
lastfm	1	0.8	2	10	top	0.5	40	0.868090395	0.991952	0.05	TV	0.05
lastfm	1	0.8	2	10	top	0.5	40	0.8445712	0.989022	0.1	TV	0.1
lastfm	1	0.8	2	10	top	0.5	40	0.844456853	0.987365	0.15	TV	0.15
lastfm	1	0.8	2	10	top	0.5	40	0.784518845	0.972931	0.2	TV	0.2
lastfm	1	0.8	2	10	top	0.5	40	0.774171265	0.977508	0.25	TV	0.25
lastfm	1	0.8	2	10	top	0.5	40	0.720468134	0.948817	0.3	TV	0.3

Figure 4.22: Arithmetic Results

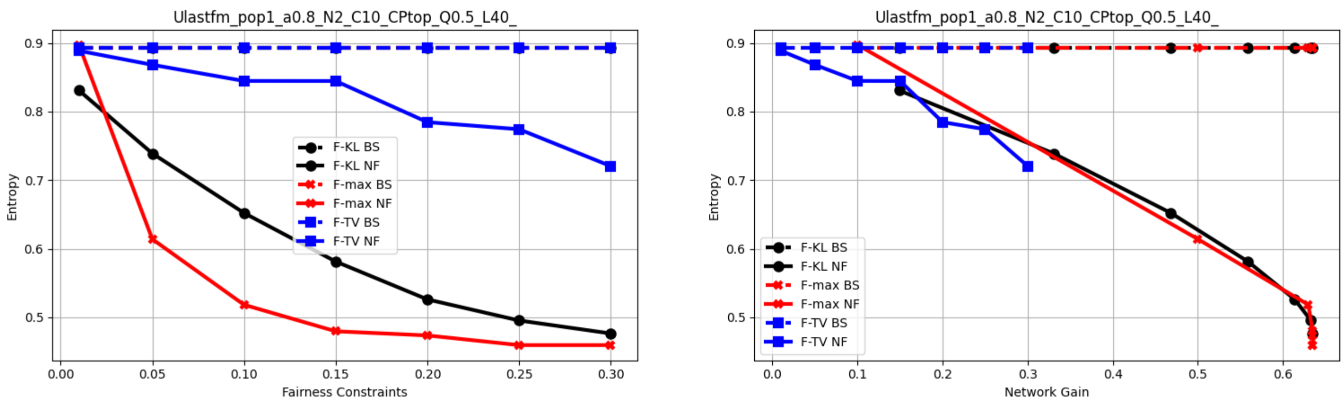

Figure 4.23: Entropy in relation with Fairness Constraints and Network Gain

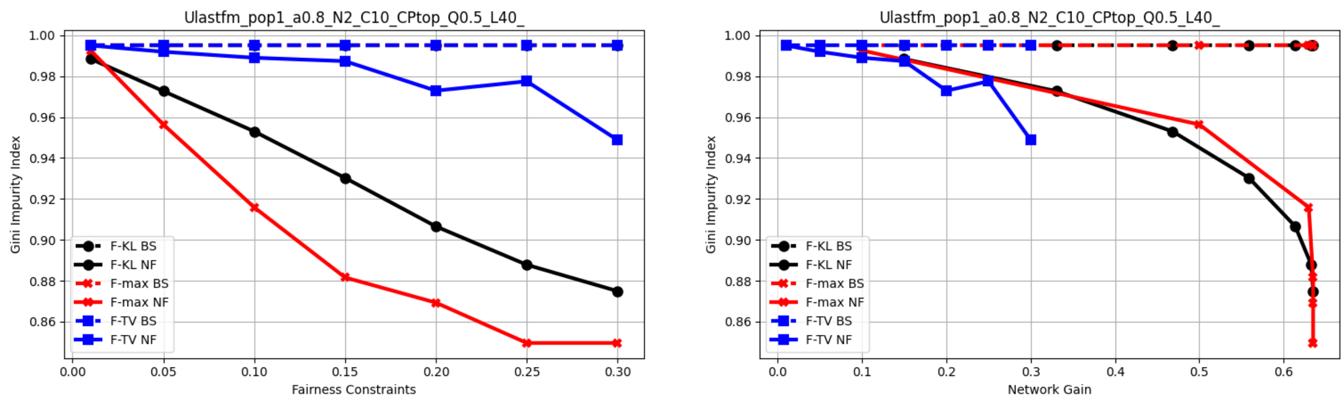

Figure 4.24: Gini Index in relation with Fairness Constraints and Network Gain

Observations 6

- Range of constraints $\in [0, 0.3]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results.
  Fairness Constraints $\uparrow ----> Gini, Entropy \downarrow$
  While F_{tv} appears to have a higher entropy than the other two metrics, when taking into account network gains, F_{tv} reaches a maximum of 50% in comparison to the gains of the other functions.
- Network Gains $\uparrow ----> Gini, Entropy \downarrow$

7. **MovieLens dataset with parameters pop=0,a=0.99, N=5, C=20, q=0.5, L=40**

Now we have changed the dataset from LastFM to MovieLens.

entropy_BS	gini_BS
0.908408	0.997462

Figure 4.25: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
movielens1k	0	0.99	5	20	top	0.5	40	0.835154321	0.993932	0.177899408	KL	0.01
movielens1k	0	0.99	5	20	top	0.5	40	0.755121585	0.986885	0.362011716	KL	0.05
movielens1k	0	0.99	5	20	top	0.5	40	0.677576367	0.978493	0.512031486	KL	0.1
movielens1k	0	0.99	5	20	top	0.5	40	0.613729327	0.970972	0.619232236	KL	0.15
movielens1k	0	0.99	5	20	top	0.5	40	0.56055768	0.964226	0.699173112	KL	0.2
movielens1k	0	0.99	5	20	top	0.5	40	0.517127457	0.958451	0.760065815	KL	0.25
movielens1k	0	0.99	5	20	top	0.5	40	0.482871905	0.95356	0.804976214	KL	0.3
movielens1k	0	0.99	5	20	top	0.5	40	0.888841556	0.993956	0.2	max	0.01
movielens1k	0	0.99	5	20	top	0.5	40	0.442238223	0.94865	0.8552873	max	0.05
movielens1k	0	0.99	5	20	top	0.5	40	0.435601818	0.944075	0.8552873	max	0.1
movielens1k	0	0.99	5	20	top	0.5	40	0.438669078	0.946014	0.8552873	max	0.15
movielens1k	0	0.99	5	20	top	0.5	40	0.43780662	0.945332	0.8552873	max	0.2
movielens1k	0	0.99	5	20	top	0.5	40	0.43866404	0.946014	0.8552873	max	0.25
movielens1k	0	0.99	5	20	top	0.5	40	0.440560446	0.947512	0.855287301	max	0.3
movielens1k	0	0.99	5	20	top	0.5	40	0.90655326	0.997365	0.01	TV	0.01
movielens1k	0	0.99	5	20	top	0.5	40	0.897287853	0.99683	0.05	TV	0.05
movielens1k	0	0.99	5	20	top	0.5	40	0.888507856	0.996128	0.1	TV	0.1
movielens1k	0	0.99	5	20	top	0.5	40	0.870794426	0.994879	0.15	TV	0.15
movielens1k	0	0.99	5	20	top	0.5	40	0.851788404	0.993298	0.2	TV	0.2
movielens1k	0	0.99	5	20	top	0.5	40	0.811613414	0.988519	0.25	TV	0.25
movielens1k	0	0.99	5	20	top	0.5	40	0.806568076	0.988406	0.3	TV	0.3

Figure 4.26: Arithmetic Results

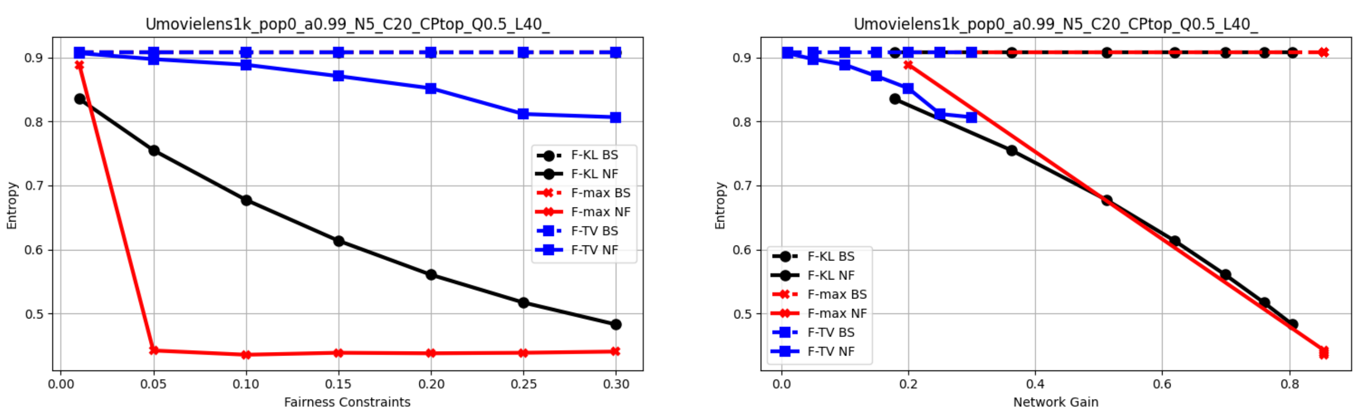

Figure 4.27: Entropy in relation with Fairness Constraints and Network Gain

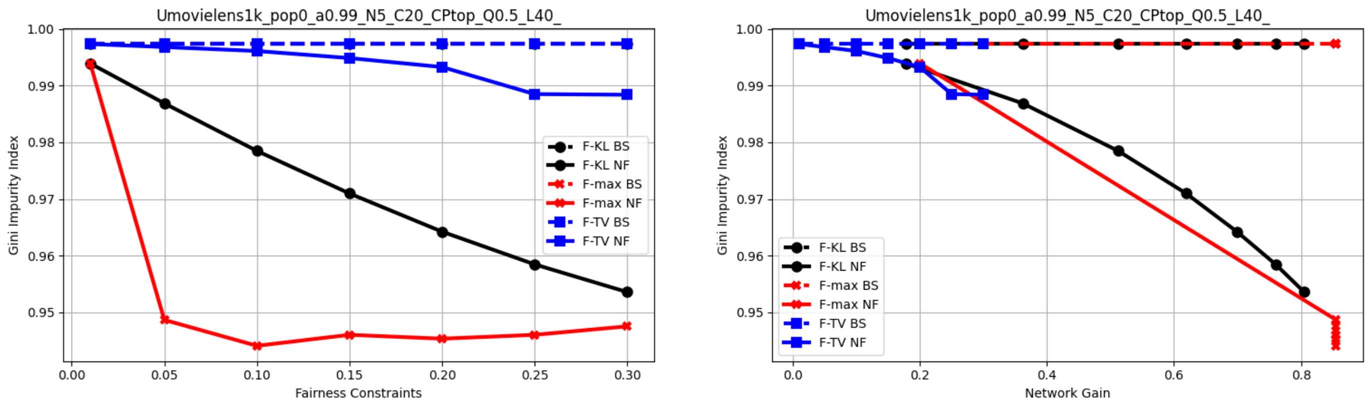

Figure 4.28: Gini Index in relation with Fairness Constraints and Network Gain

Observations 7

- Range of constraints $\in [0, 0.3]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results.
  Fairness Constraints $\uparrow ----> Gini, Entropy \downarrow$
  F_{tv} closer to the bounds of Baseline RS in comparison with the other 2 functions. Especially for low threshold of unfairness , NF coincides with BS.
- Network Gains $\uparrow ----> Gini, Entropy \downarrow$ Entropy may be in general high but network gains are really low in F_{tv} .

8. **MovieLens dataset with parameters pop=0, a=0.99, N=5, C=20, q=0.8, L=40**

In this scenario, we increase the q constraint:
q(0.5 $\rightarrow$ 0.8)

By increasing the quality constraint, we observe a smoother decrease of entropy compared to the previous scenario where it was a lot steeper.

entropy_BS	gini_BS
0.908408	0.997462

Figure 4.29: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
movielens1k	0	0.99	5	20	top	0.8	40	0.836480476	0.993969	0.176700578	KL	0.01
movielens1k	0	0.99	5	20	top	0.8	40	0.756616545	0.98694	0.36086489	KL	0.05
movielens1k	0	0.99	5	20	top	0.8	40	0.679003971	0.978358	0.510400419	KL	0.1
movielens1k	0	0.99	5	20	top	0.8	40	0.614688121	0.970522	0.616752872	KL	0.15
movielens1k	0	0.99	5	20	top	0.8	40	0.562003972	0.963673	0.696220501	KL	0.2
movielens1k	0	0.99	5	20	top	0.8	40	0.517103317	0.956499	0.756041326	KL	0.25
movielens1k	0	0.99	5	20	top	0.8	40	0.478892338	0.948289	0.798551018	KL	0.3
movielens1k	0	0.99	5	20	top	0.8	40	0.848802136	0.993433	0.2	max	0.01
movielens1k	0	0.99	5	20	top	0.8	40	0.462469078	0.950013	0.816031415	max	0.05
movielens1k	0	0.99	5	20	top	0.8	40	0.390074121	0.912045	0.842302287	max	0.1
movielens1k	0	0.99	5	20	top	0.8	40	0.347712425	0.879031	0.842493947	max	0.15
movielens1k	0	0.99	5	20	top	0.8	40	0.347879955	0.879309	0.842497039	max	0.2
movielens1k	0	0.99	5	20	top	0.8	40	0.347690003	0.878988	0.842497039	max	0.25
movielens1k	0	0.99	5	20	top	0.8	40	0.347814471	0.8792	0.842497078	max	0.3
movielens1k	0	0.99	5	20	top	0.8	40	0.906986534	0.99739	0.01	TV	0.01
movielens1k	0	0.99	5	20	top	0.8	40	0.893104612	0.996743	0.05	TV	0.05
movielens1k	0	0.99	5	20	top	0.8	40	0.887416931	0.996111	0.1	TV	0.1
movielens1k	0	0.99	5	20	top	0.8	40	0.874807222	0.994938	0.15	TV	0.15
movielens1k	0	0.99	5	20	top	0.8	40	0.857079748	0.993409	0.2	TV	0.2
movielens1k	0	0.99	5	20	top	0.8	40	0.792451079	0.98922	0.25	TV	0.25
movielens1k	0	0.99	5	20	top	0.8	40	0.812205028	0.988404	0.3	TV	0.3

Figure 4.30: Arithmetic Results

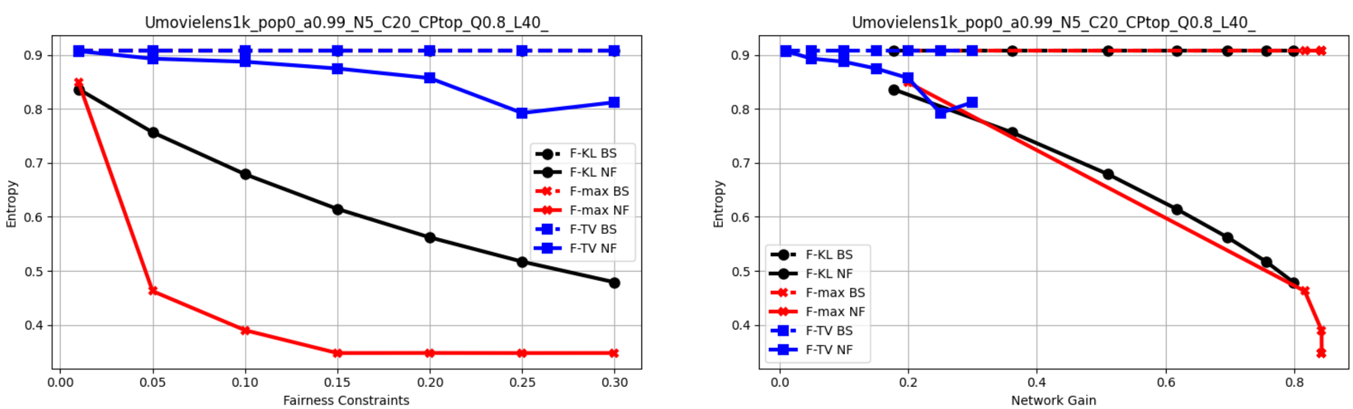

Figure 4.31: Entropy in relation with Fairness Constraints and Network Gain

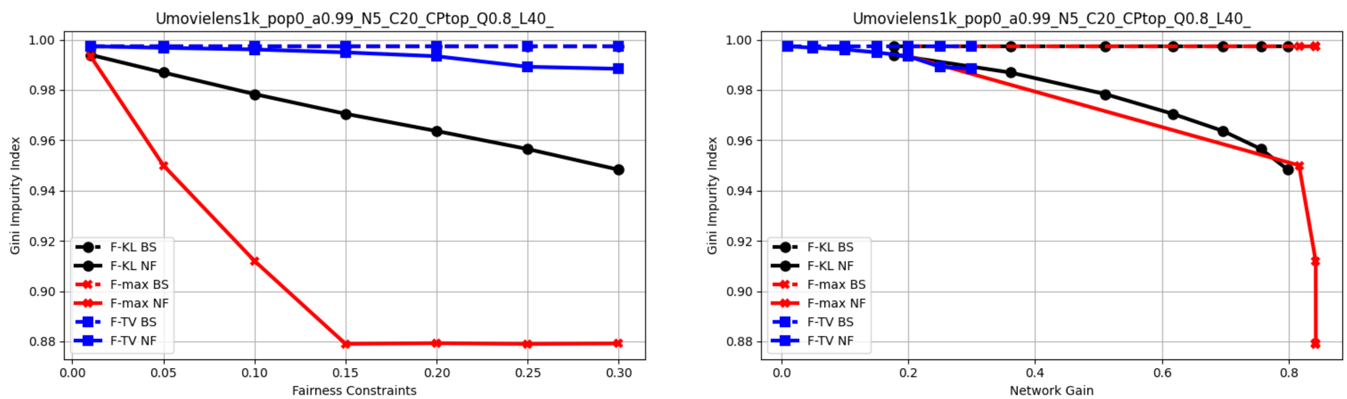

Figure 4.32: Gini Index in relation with Fairness Constraints and Network Gain

Observations 8

- Range of constraints $\in [0, 0.3]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results.
  Fairness Constraints $\uparrow - - -- > Gini, Entropy \downarrow$
  F_{tv} closer to the bounds of Baseline RS in comparison with the other 2 functions. Especially for low threshold of unfairness , NF aligns with BS.
- Network Gains $\uparrow - - -- > Gini, Entropy \downarrow$ in F_{tv} .

9. **MovieLens dataset with parameters pop=1, a=0.8, N=2, C=10,q=0.5,L=40**

In this scenario, there are more aggressive changes in the attributes:

N(5 $\rightarrow$ 2) , q(0.8 $\rightarrow$ 0.5),pop(0 $\rightarrow$ 1) , a(0.99 $\rightarrow$ 0.8), C(20 $\rightarrow$ 10)

The size of the recommendation list and the Cache size are reduced. Popularity has changed to "1" and the q constraint has decreased, while also the value of probability "a" has dropped. This leads to a reduced recommendation list with reduced recommendation quality compared to the previous scenario, as well as a smaller cache size resulting in a lower number of cache selections. The items are now popular due to the change in their popularity. All the above contribute to recommendations that lack variety and both the metrics verify that as they achieve lower overall scores for every fairness constraint. Furthermore, baseline values are also lower than before.

entropy_BS	gini_BS
0.856184	0.994887

Figure 4.33: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
movielens1k	1	0.8	2	10	top	0.5	40	0.776691849	0.985471	0.179679476	KL	0.01
movielens1k	1	0.8	2	10	top	0.5	40	0.691795378	0.969018	0.350381402	KL	0.05
movielens1k	1	0.8	2	10	top	0.5	40	0.613991321	0.950814	0.482700403	KL	0.1
movielens1k	1	0.8	2	10	top	0.5	40	0.557562227	0.934554	0.567534052	KL	0.15
movielens1k	1	0.8	2	10	top	0.5	40	0.517605804	0.917017	0.614317513	KL	0.2
movielens1k	1	0.8	2	10	top	0.5	40	0.494441052	0.89932	0.627599588	KL	0.25
movielens1k	1	0.8	2	10	top	0.5	40	0.485378098	0.904543	0.62957843	KL	0.3
movielens1k	1	0.8	2	10	top	0.5	40	0.85224015	0.991958	0.1	max	0.01
movielens1k	1	0.8	2	10	top	0.5	40	0.605751639	0.956765	0.5	max	0.05
movielens1k	1	0.8	2	10	top	0.5	40	0.497650228	0.91746	0.629366385	max	0.1
movielens1k	1	0.8	2	10	top	0.5	40	0.484014991	0.901412	0.629576951	max	0.15
movielens1k	1	0.8	2	10	top	0.5	40	0.480623017	0.895406	0.629576921	max	0.2
movielens1k	1	0.8	2	10	top	0.5	40	0.430783219	0.83555	0.629576953	max	0.25
movielens1k	1	0.8	2	10	top	0.5	40	0.446304631	0.854368	0.629578237	max	0.3
movielens1k	1	0.8	2	10	top	0.5	40	0.851784236	0.994627	0.01	TV	0.01
movielens1k	1	0.8	2	10	top	0.5	40	0.845540191	0.993875	0.05	TV	0.05
movielens1k	1	0.8	2	10	top	0.5	40	0.835059848	0.992052	0.1	TV	0.1
movielens1k	1	0.8	2	10	top	0.5	40	0.816373811	0.988822	0.15	TV	0.15
movielens1k	1	0.8	2	10	top	0.5	40	0.796158664	0.986209	0.2	TV	0.2
movielens1k	1	0.8	2	10	top	0.5	40	0.770543493	0.977894	0.25	TV	0.25
movielens1k	1	0.8	2	10	top	0.5	40	0.745876465	0.973752	0.3	TV	0.3

Figure 4.34: Arithmetic Results

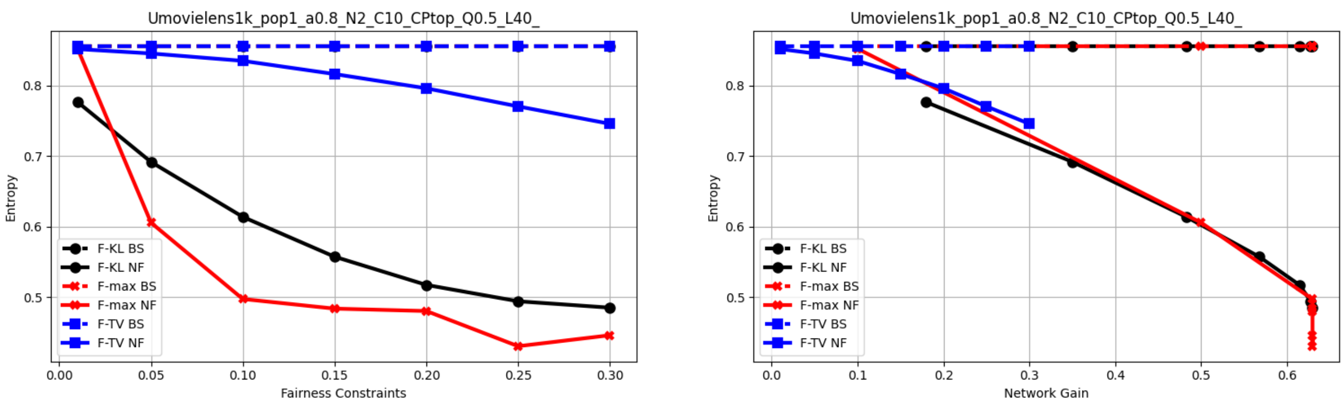

Figure 4.35: Entropy in relation with Fairness Constraints and Network Gain

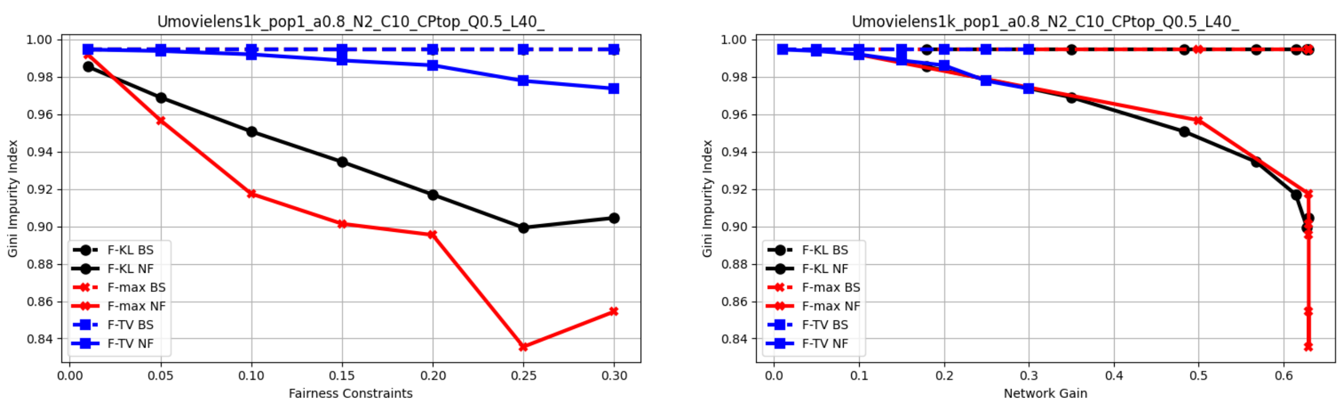

Figure 4.36: Gini Index in relation with Fairness Constraints and Network Gain

Observations 9

- Range of constraints $\in [0, 0.3]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results.
  Fairness Constraints $\uparrow ----> Gini, Entropy \downarrow$
  F_{tv} closer to the bounds of Baseline RS
- Network Gains $\uparrow ----> Gini, Entropy \downarrow$ in F_{tv} .

10. **MovieLens dataset with parameters pop=1, a=0.99, N=2, C=5, q=0.9, L=40**

In this scenario we increase the q constraint and the probability of a user to follow the recommendation but decrease the Cache size:
q(0.5 $\rightarrow$ 0.9), a(0.8 $\rightarrow$ 0.99), C(10 $\rightarrow$ 5)

Although there is an increase in the quality constraint and the probability of a user to follow the recommended item is higher, the decrease in Cache size by half while having a small recommendation list (N=2), leads to an overall drop in entropy and Gini. Even though NF entropy appears to be higher than BS for a single value of unfairness, the graph of NF entropy significantly deviates from BS in comparison to the previous scenario.

entropy_BS	gini_BS
0.774481	0.992104

Figure 4.37: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
movielens1k	1	0.99	2	5	top	0.9	40	0.695605447	0.979023	0.171789101	KL	0.01
movielens1k	1	0.99	2	5	top	0.9	40	0.610402501	0.951365	0.336334408	KL	0.05
movielens1k	1	0.99	2	5	top	0.9	40	0.469730298	0.883649	0.519962465	KL	0.15
movielens1k	1	0.99	2	5	top	0.9	40	0.344403616	0.811242	0.63656853	KL	0.3
movielens1k	1	0.99	2	5	top	0.9	40	0.286940207	0.780169	0.67624982	KL	0.45
movielens1k	1	0.99	2	5	top	0.9	40	0.270048188	0.773113	0.682366503	KL	0.6
movielens1k	1	0.99	2	5	top	0.9	40	0.856226787	0.992252	0.05	max	0.01
movielens1k	1	0.99	2	5	top	0.9	40	0.672585991	0.967104	0.25	max	0.05
movielens1k	1	0.99	2	5	top	0.9	40	0.416881195	0.919496	0.454046893	max	0.1
movielens1k	1	0.99	2	5	top	0.9	40	0.381598379	0.886619	0.533616112	max	0.15
movielens1k	1	0.99	2	5	top	0.9	40	0.345840129	0.839534	0.607378469	max	0.2
movielens1k	1	0.99	2	5	top	0.9	40	0.27819912	0.782246	0.670602241	max	0.25
movielens1k	1	0.99	2	5	top	0.9	40	0.771088629	0.991751	0.01	TV	0.01
movielens1k	1	0.99	2	5	top	0.9	40	0.746737681	0.985895	0.1	TV	0.1
movielens1k	1	0.99	2	5	top	0.9	40	0.700944532	0.971835	0.2	TV	0.2
movielens1k	1	0.99	2	5	top	0.9	40	0.536582828	0.936396	0.4000036	TV	0.4
movielens1k	1	0.99	2	5	top	0.9	40	0.415104965	0.870059	0.558171466	TV	0.6
movielens1k	1	0.99	2	5	top	0.9	40	0.303427115	0.790484	0.667416439	TV	0.8

Figure 4.38: Arithmetic Results

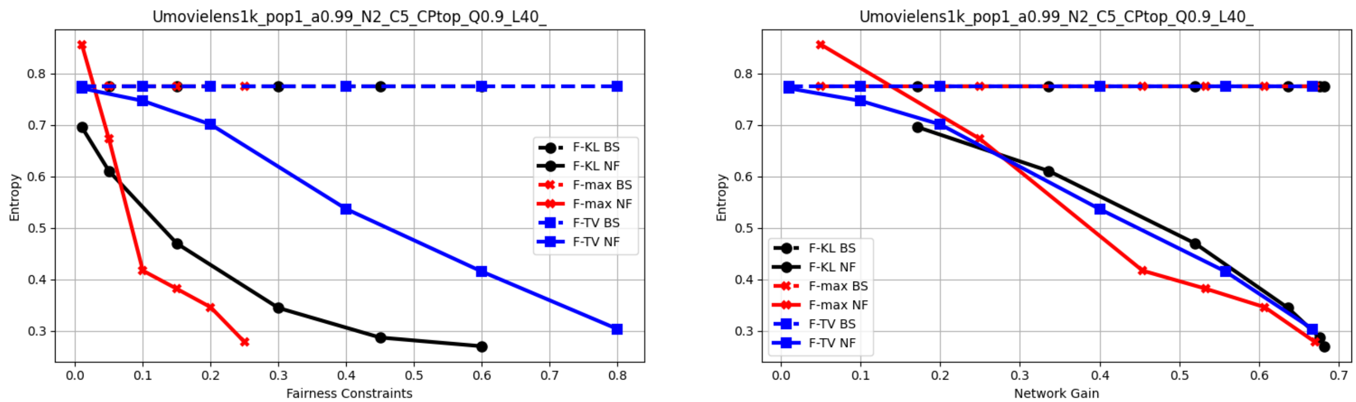

Figure 4.39: Entropy in relation with Fairness Constraints and Network Gain

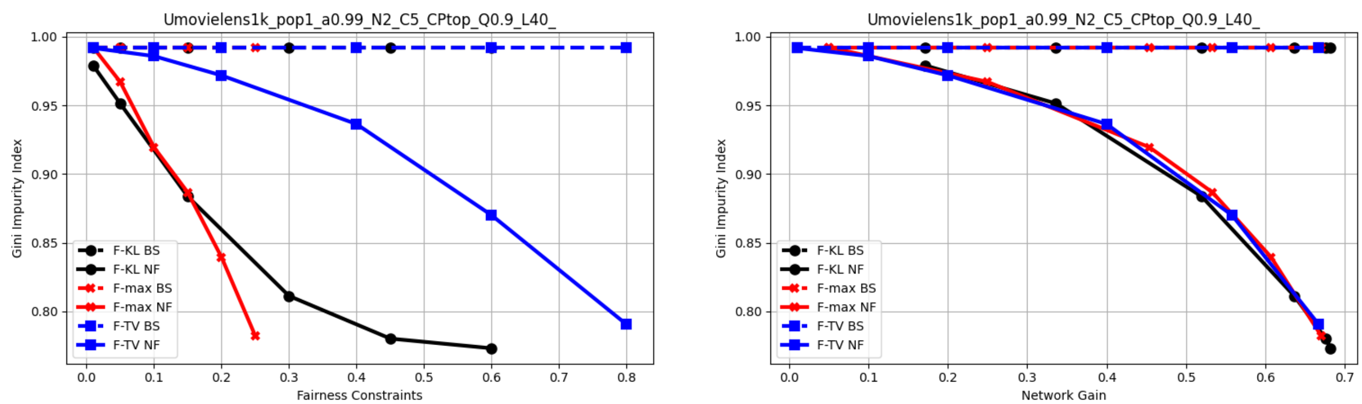

Figure 4.40: Gini Index in relation with Fairness Constraints and Network Gain

Observations 10

- Range of constraints $\in [0, 0.8]$
- Gini IMID and Entropy agree on the results.
  Fairness Constraints $\uparrow - - -- > Gini, Entropy \downarrow$
  A small spike of F_{max} is observed, particularly in Entropy of NF, where for a low value of fairness constraints the Entropy seems to be **above the Baseline's** . Since this is only one sample, it cannot be taken into serious considerations. Additionally, the F_{max} graph steepens and drops dramatically after that spike, providing evidence that coincides with the previous assumptions that F_{tv} is the closest to Baseline.
- Network Gains $\uparrow - - -- > Gini, Entropy \downarrow$

11. **MovieLens dataset with parameters pop=1, a=0.99, N=2, C=10, q=0.8, L=40**

In this scenario, we increase the Cache size and decrease the q constraint:
q(0.9 $\rightarrow$ 0.8), C(5 $\rightarrow$ 10)

Although the q constraint experiences a slight decrease, there is an increase in cache size. The results are similar with the previous scenario.
It is important to acknowledge that the previous scenario covers a wider range of fairness constraints compared to this one, which is why the graphical differences between the two scenarios are noticeable.

entropy_BS	gini_BS
0.774481	0.992104

Figure 4.41: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
movielens1k	1	0.99	2	10	top	0.8	40	0.69203714	0.981424	0.195429841	KL	0.01
movielens1k	1	0.99	2	10	top	0.8	40	0.615258844	0.964836	0.370352911	KL	0.05
movielens1k	1	0.99	2	10	top	0.8	40	0.545341717	0.94731	0.504069316	KL	0.1
movielens1k	1	0.99	2	10	top	0.8	40	0.488884155	0.930578	0.59380332	KL	0.15
movielens1k	1	0.99	2	10	top	0.8	40	0.442755587	0.913273	0.655803874	KL	0.2
movielens1k	1	0.99	2	10	top	0.8	40	0.405091879	0.8974	0.699589245	KL	0.25
movielens1k	1	0.99	2	10	top	0.8	40	0.372974415	0.880304	0.730246328	KL	0.3
movielens1k	1	0.99	2	10	top	0.8	40	0.842154208	0.990156	0.1	max	0.01
movielens1k	1	0.99	2	10	top	0.8	40	0.562412896	0.95065	0.5	max	0.05
movielens1k	1	0.99	2	10	top	0.8	40	0.355409174	0.895305	0.756117249	max	0.1
movielens1k	1	0.99	2	10	top	0.8	40	0.313329056	0.852725	0.771544316	max	0.15
movielens1k	1	0.99	2	10	top	0.8	40	0.301035296	0.829442	0.774372605	max	0.2
movielens1k	1	0.99	2	10	top	0.8	40	0.271602018	0.776325	0.777200894	max	0.25
movielens1k	1	0.99	2	10	top	0.8	40	0.221520017	0.709918	0.779503914	max	0.3
movielens1k	1	0.99	2	10	top	0.8	40	0.770674048	0.991767	0.01	TV	0.01
movielens1k	1	0.99	2	10	top	0.8	40	0.765254149	0.990598	0.05	TV	0.05
movielens1k	1	0.99	2	10	top	0.8	40	0.75195542	0.988581	0.1	TV	0.1
movielens1k	1	0.99	2	10	top	0.8	40	0.709924309	0.980571	0.150002856	TV	0.15
movielens1k	1	0.99	2	10	top	0.8	40	0.704817442	0.978833	0.2	TV	0.2
movielens1k	1	0.99	2	10	top	0.8	40	0.643884639	0.956391	0.250002904	TV	0.25
movielens1k	1	0.99	2	10	top	0.8	40	0.677338798	0.970585	0.3	TV	0.3

Figure 4.42: Arithmetic Results

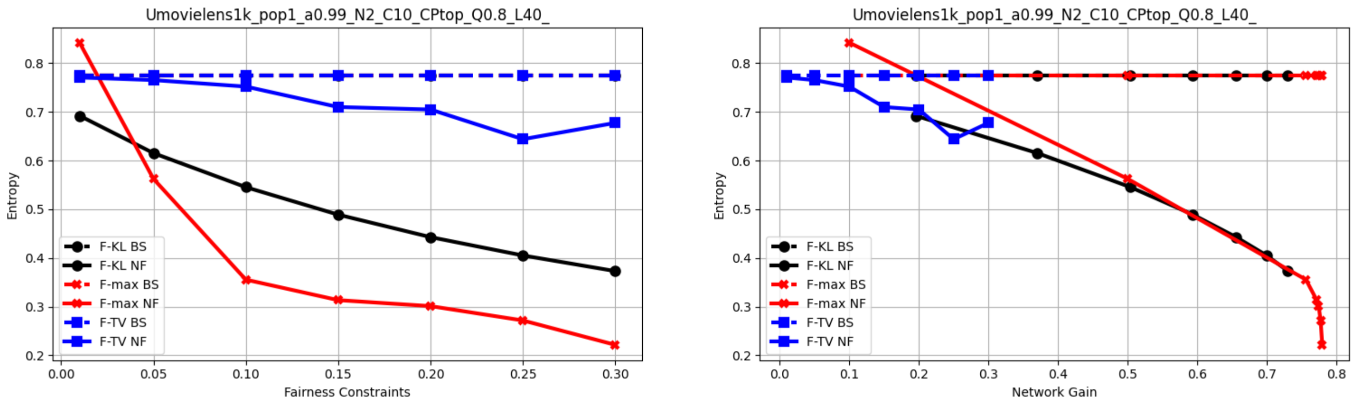

Figure 4.43: Entropy in relation with Fairness Constraints and Network Gain

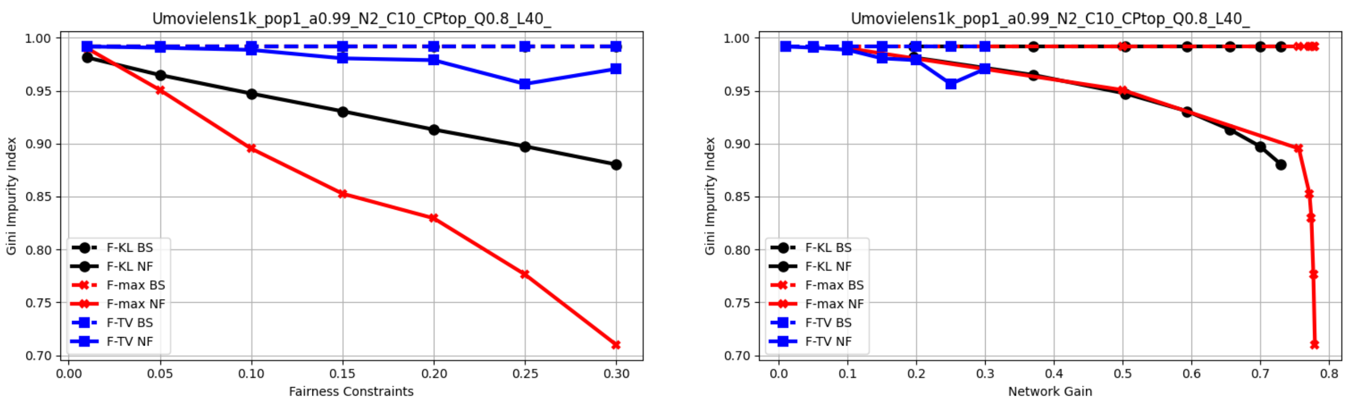

Figure 4.44: Gini Index in relation with Fairness Constraints and Network Gain

Observations 11

- Range of constraints $\in [0, 0.3]$ where F_{tv} is closer to the bounds of Baseline recommendations.
- Gini IMID and Entropy agree on the results.
  Fairness Constraints $\uparrow - - - - > Gini, Entropy \downarrow$
  There is a slight increase, particularly in the measure of entropy, under the given scenario. Following that particular example, the plots exhibit a greater degree of familiarity with the previous ones. (F_{tv} is the closest to the Baseline)
- Network Gains $\uparrow - - - - > Gini, Entropy \downarrow$

12. **MovieLens dataset with parameters pop=1, a=0.99, N=2, C=10, q=0.9, L=40**

In this scenario we increase the q constraint:

q(0.8 → 0.9)

By improving the q constraint there is a small increase in overall Entropy values for this range of fairness constraints.

entropy_BS	gini_BS
0.774481	0.992104

Figure 4.45: Baseline values of Metrics

U	pop	alpha	N	C	cp	q	L	entropy_NF	gini_NF	gain	F	constr
movielens1k	1	0.99	2	10	top	0.9	40	0.693646381	0.981537	0.194285366	KL	0.01
movielens1k	1	0.99	2	10	top	0.9	40	0.614631979	0.963711	0.367303062	KL	0.05
movielens1k	1	0.99	2	10	top	0.9	40	0.492866546	0.92997	0.550410379	KL	0.15
movielens1k	1	0.99	2	10	top	0.9	40	0.389597844	0.87657	0.629304582	KL	0.3
movielens1k	1	0.99	2	10	top	0.9	40	0.326515408	0.818661	0.646103829	KL	0.4
movielens1k	1	0.99	2	10	top	0.9	40	0.299869363	0.79488	0.649000846	KL	0.45
movielens1k	1	0.99	2	10	top	0.9	40	0.287992782	0.785422	0.64959375	KL	0.6
movielens1k	1	0.99	2	10	top	0.9	40	0.813275524	0.989224	0.1	max	0.01
movielens1k	1	0.99	2	10	top	0.9	40	0.533616644	0.948181	0.5	max	0.05
movielens1k	1	0.99	2	10	top	0.9	40	0.405587955	0.918077	0.62887328	max	0.1
movielens1k	1	0.99	2	10	top	0.9	40	0.371968629	0.887818	0.64190214	max	0.15
movielens1k	1	0.99	2	10	top	0.9	40	0.355147085	0.864125	0.645216229	max	0.2
movielens1k	1	0.99	2	10	top	0.9	40	0.321536891	0.818879	0.648243698	max	0.25
movielens1k	1	0.99	2	10	top	0.9	40	0.769832919	0.99171	0.010006816	TV	0.01
movielens1k	1	0.99	2	10	top	0.9	40	0.755629929	0.988461	0.1	TV	0.1
movielens1k	1	0.99	2	10	top	0.9	40	0.721430174	0.980717	0.2	TV	0.2
movielens1k	1	0.99	2	10	top	0.9	40	0.563141271	0.948742	0.400002835	TV	0.4
movielens1k	1	0.99	2	10	top	0.9	40	0.447878247	0.926239	0.598128435	TV	0.6
movielens1k	1	0.99	2	10	top	0.9	40	0.307759599	0.803658	0.649016839	TV	0.8

Figure 4.46: Arithmetic Results

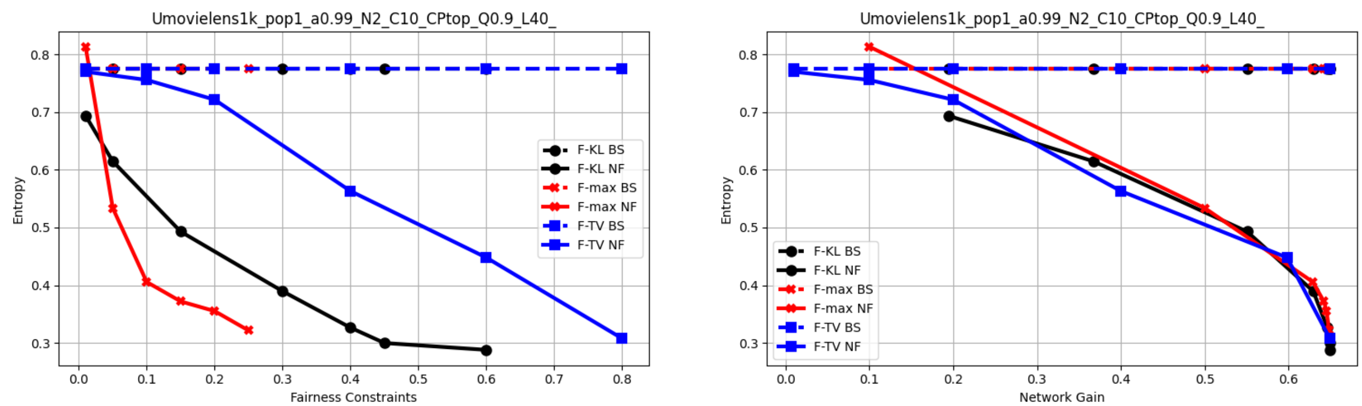

Figure 4.47: Entropy in relation with Fairness Constraints and Network Gain

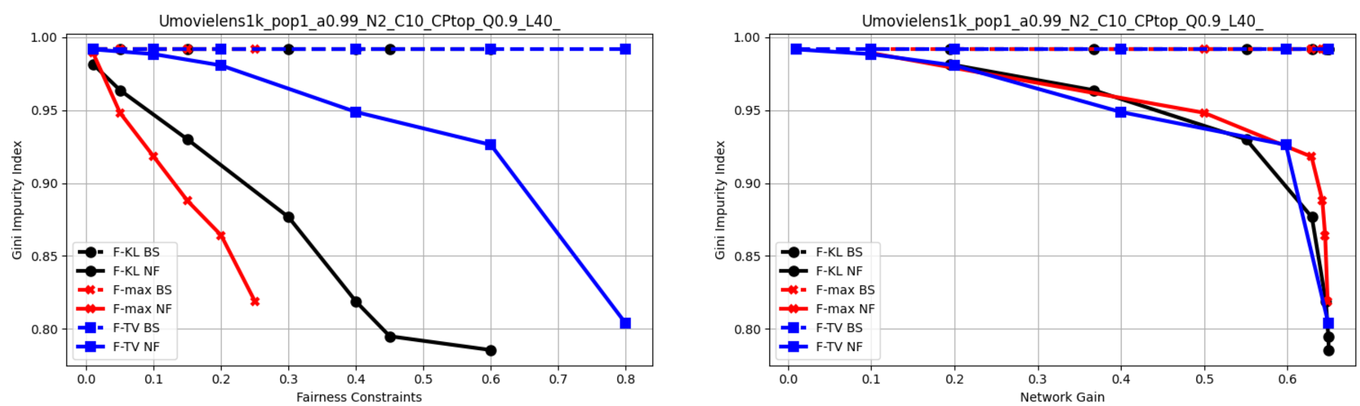

Figure 4.48: Gini Index in relation with Fairness Constraints and Network Gain

Observations 12

- Range of constraints $\in [0, 0.8]$
- Gini IMID and Entropy agree on the results.
  Same behavior as the previous 2 Scenarios. Fairness Constraints $\uparrow ---- > Gini, Entropy \downarrow$
- Network Gains $\uparrow ---- > Gini, Entropy \downarrow$
  Plots illustrating network gains of NF deviate significantly from the Baseline RS.

The problem with Variance

Before we proceed to the final conclusions, we need to explain why we have not used the third metric that we proposed. Unfortunately, *Variance* could not help us investigate the problem of Content bubble and we are about to explain the reasons why.

Entropy is our fundamental measure because it quantifies uncertainty, which directly relates to variety. Variance would be used as a complementary metric that would be compared to the results of Entropy.

After processing the data and generating the first plots, including all the metrics (Entropy,Gini and Variance) we observed some irregularities as to what we expected with variance as a measure of diversity. Although Entropy and Gini had similar results (as presented before), variance seemed to have a different behavior. More specifically, as Gini and Entropy values were decreasing, variance seemed to increase. Low values of Gini and Entropy signify the absence of diversity between the recommendations. Such a scenario might indicate that while the probabilities are spread out, they are not spread out evenly across items, leading to low entropy.

Low entropy and high variance in recommendation system probability vectors indicate that although there is variability in the probability values assigned to different items, the overall distribution remains skewed, with certain items having significantly higher probabilities than others. The problem arises when the recommendation algorithm favors a small number of popular items with high probabilities while giving lower probabilities to a greater number of less popular items. This leads to a recommendation list that lacks diversity, despite having a wide range of probability values.

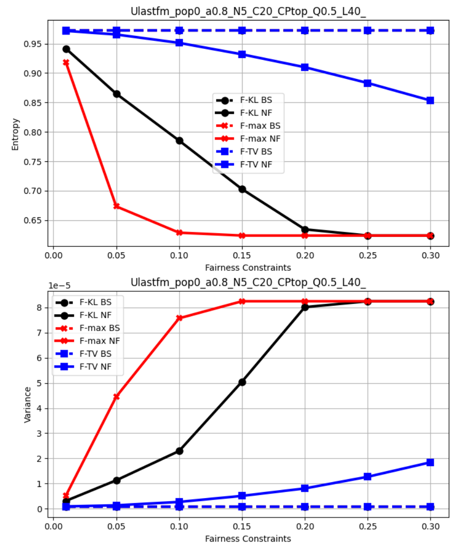

Figure 4.49: Comparing Entropy and Variance

4.10 Overview

The outcomes of our study have clearly shown that the Fair-NFRS algorithm may be useful in terms of managing fairness, but this advantage is accompanied by a significant drawback: the algorithm generates recommendations with low entropy, indicating a lack of diversity in the content suggested. The Gini Impurity Index confirms the results for entropy and demonstrates that the given recommendations yield low levels of Gini Index, indicating a uniformly distributed and low impurity with minimal variation.

The three Functions

Furthermore, when examining the graphs representing various fairness metrics, it is evident that F_{max}, which measures individual fairness, has lower values. Conversely, F_{tv}, which is calculated by averaging across all elements, has the largest values of entropy and Gini IMID, reaching the boundaries of baseline Rs. F_{kl} falls within the range of the other two methods.

The disparities in curvature between the two metrics may arise from the inherent characteristics of their algorithms. The concept of entropy incorporates the calculation of probabilities squared, while the Gini impurity index includes a logarithmic term.

Summary

These points summarize the above results :

- The Fair-NFRS will generate recommendations with limited diversity, as indicated by the two metrics. This will result in a more limited user experience in terms of content variety.
- The Fair-NFRS exhibits high levels of Entropy by permitting little instances of unfairness, occasionally corresponding with the limits of Baseline RS.
- The low scores of Network Gain overwhelm the high values of Entropy and Gini impurity index.
- As the values of Fairness Constraints increase, the values of Entropy and Gini decrease.The maximum scores in network gains are accompanied by the minimum values of the Metrics.

Chapter 5

Conclusions

The primary objective of this thesis was to make a contribution to the existing research on "Fairness in Network Friendly Recommendations" as outlined in the publication by [2]. This work analyzes how some NFR algorithms affect the recommendation in comparison with the baseline system. These algorithms are said to enhance network gains, but they also bring to light the issue of unfairness in the recommended items. In order to address the problem, the researchers suggested implementing a Fair-NFR strategy that is both conscious of fairness and establishes specific fairness constraints to regulate the unfairness of the system. The proposed algorithm seemed to provided good results about fairness and with the cost of allowing some unfairness ,achieved in many cases higher network gains than some NFR algorithms without fairness.

Meeting the Expectations

The necessity of confirming the value of fairness and network advantages presented in the preceding results became the next challenge. Essentially, the primary goal of recommendation systems is to *offer suggestions that enhance the overall user experience*. The fairness aspect addresses the concerns of content owners/producers that were previously overlooked in favor of prioritizing user wants.Furthermore, FAIR-NFR significantly improves network benefits, resulting in an enhanced user experience through the delivery of high-quality information and reduced network latencies.

Another technique must be taken to investigate these two advantages, which is to ensure that recommendations also captivate the user's interest by offering information that is diverse and varied.Providing users with a wider array of choices will help maintain their interest in the service and prevent them from experiencing a lack of novelty or serendipity.

Functionality of the Metrics

The two metrics that we proposed measure the diversity and the variety of the recommendations. Entropy could help assess the diversity of recommended items or the level of surprise in recommendations for users and Gini IMID might be used to measure the uncertainty or impurity of the distribution .

In Chapter 4, we demonstrated that both of these metrics indicate that allowing a higher level of unfairness leads to significant network gains, although at the cost of reduced suggestion diversity. Entropy and Gini exhibit high values when there is some tolerance for unfairness. However, in this particular scenario, the network gains are very low. It is important to note that the entire study is focused on Recommendation systems that are designed to be Network friendly and prioritize network benefits. The technique under consideration is the *Multi-step NFRS* which, in general, generates greater network gains but on the other hand leads to less fairness, as stated in the paper.

Additionally, out of the three Fairness functions, F_{tv} is the most efficient due to the fact that its results are the closest to the boundaries of the Baseline RS.

An Epilogue

The big picture is that by prioritizing unfairness to maximize network advantages, there is a trade-off in terms of sacrificing entropy. The network provider will determine the level of tolerance for unfairness based on their specific requirements. The concepts of network gain, fairness, and entropy/Gini IMID all have significant importance in the context of recommendations and should be thoroughly taken into account. The trade-offs between unfairness measured by Entropy/Gini and the improvements in network measured by Entropy/Gini-Network Gain have demonstrated their high sensitivity to changes, highlighting the complexity of this topic.

5.1 Future Work

This thesis examined introduced Entropy and Gini IMID in order to measure the variety and the diversity of the produced recommendations under a Network-frinedly protocol with fairness constraints.

Content Bubble

The initial observations of the outcomes were intriguing regarding this innovative topic in recommendation systems. The diversity and variety of the recommendations are closely linked to the issue of the *Content Bubble* which illustrates the phenomenon where users are mostly exposed to content that aligns with their interests, but their exposure to diverse and varied content is restricted. This can result in a constricted viewpoint for the user and a strengthening of preexisting preferences or biases.

The presence of low entropy contributes to the issue of *Content Bubble*, making it a compelling subject to explore methods of increasing entropy in order to eliminate this phenomenon.

Another Approach

In this instance, we examined a particular situation with specific variables outlined in 4.7. A more sensitive analysis could potentially provide a more valuable understanding of the issue of Entropy and network gains, particularly when considering variables with multiple values that are examined individually. Definitions such as quality of recommendations (QoR) should be utilized and thoroughly analyzed as they can offer valuable insights into fairness and the impact of entropy on the quality of recommended content.

When the quality of recommendation (QoR) is relaxed, the network-friendly recommendation system (NF-RS) gains greater freedom to modify the recommendation lists. As a result, this pushes the content demand p^{NF} further away from p^{BS}.

Bibliography

[1] L. Vigneri, T. Spyropoulos, and C. Barakat, "Low cost video streaming through mobile edge caching: Modelling and optimization," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1302–1315, 2018.

[2] T. Giannakas, P. Sermpezis, A. Giovanidis, T. Spyropoulos, and G. Arvanitakis, "Fairness in network-friendly recommendations," in *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 71–80, IEEE, 2021.

[3] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[4] J. Deng, Y. Wang, J. Guo, Y. Deng, J. Gao, and Y. Park, "A similarity measure based on kullback–leibler divergence for collaborative filtering in sparse data," *Journal of Information Science*, vol. 45, no. 5, pp. 656–675, 2019.

[5] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[6] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-based systems*, vol. 26, pp. 225–238, 2012.

[7] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*, vol. 30, pp. 47–56, Barcelona, 2000.

[8] Y. L. Sukestiyarno, H. A. Sapolo, and H. Sofyan, "Application of recommendation system on e-learning platform using content-based filtering with jaccard similarity and cosine similarity algorithms," 2023.

[9] P. B. Thorat, R. M. Goudar, and S. Barve, "Survey on collaborative filtering, content-based filtering and hybrid recommendation system," *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31–36, 2015.

[10] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent data analysis*, vol. 21, no. 6, pp. 1487–1524, 2017.

[11] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A hybrid approach using collaborative filtering and content based filtering for recommender system," in *Journal of Physics: Conference Series*, vol. 1000, p. 012101, IOP Publishing, 2018.

[12] D. Kluver, M. D. Ekstrand, and J. A. Konstan, "Rating-based collaborative filtering: algorithms and evaluation," *Social information access: Systems and technologies*, pp. 344–390, 2018.

[13] G. V. Kumar, P. K. Sahoo, and K. Eswaran, "A recommendation system & their performance metrics using several ml algorithms," *ResearchGate*, vol. 9, no. 3, 2020.

[14] M. M. Afsar, T. Crump, and B. Far, "Reinforcement learning based recommender systems: A survey," *ACM Computing Surveys*, vol. 55, no. 7, pp. 1–38, 2022.

[15] L. Zhang, T. Luo, F. Zhang, and Y. Wu, "A recommendation model based on deep neural network," *IEEE Access*, vol. 6, pp. 9454–9463, 2018.

[16] L. E. Molina and S. Bhulai, "Recommendation system for netflix," *Retrieved June*, vol. 6, p. 2021, 2018.

[17] F. Yu, A. Zeng, S. Gillard, and M. Medo, "Network-based recommendation algorithms: A review," *Physica A: Statistical Mechanics and its Applications*, vol. 452, pp. 192–208, 2016.

[18] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.

[19] "How tiktok recommends content| tiktok help center."

[20] M. Zhang and Y. Liu, "A commentary of tiktok recommendation algorithms in mit technology review 2021," *Fundamental Research*, vol. 1, no. 6, pp. 846–847, 2021.

[21] T. Giannakas, P. Sermpezis, and T. Spyropoulos, "Network friendly recommendations: Optimizing for long viewing sessions," *IEEE Transactions on Mobile Computing*, 2021.

[22] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Caching-aware recommendations: Nudging user preferences towards better caching performance," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.

[23] D. Tsigkari and T. Spyropoulos, "An approximation algorithm for joint caching and recommendations in cache networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1826–1841, 2022.

[24] Z. Chen, Q. He, Z. Mao, H.-M. Chung, and S. Maharjan, "A study on the characteristics of douyin short videos and implications for edge caching," in *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1–6, 2019.

[25] R. Zhou, S. Khemmarat, and L. Gao, "The impact of youtube recommendation system on video views," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 404–410, 2010.

[26] T. Giannakas, P. Sermpezis, and T. Spyropoulos, "Show me the cache: Optimizing cache-friendly recommendations for sequential content access," in *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pp. 14–22, IEEE, 2018.

[27] T. Spyropoulos and P. Sermpezis, "Soft cache hits and the impact of alternative content recommendations on mobile edge caching," in *Proceedings of the eleventh ACM workshop on challenged networks*, pp. 51–56, 2016.

[28] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Jointly optimizing content caching and recommendations in small cell networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 125–138, 2018.

[29] T. Giannakas, T. Spyropoulos, and P. Sermpezis, "The order of things: Position-aware network-friendly recommendations in long viewing sessions," in *2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, pp. 1–8, IEEE, 2019.

[30] S. Kastanakis, P. Sermpezis, V. Kotronis, and X. Dimitropoulos, "Cabaret: Leveraging recommendation systems for mobile edge caching," in *Proceedings of the 2018 Workshop on Mobile Edge Communications*, pp. 19–24, 2018.

[31] Y. Li, H. Chen, S. Xu, Y. Ge, J. Tan, S. Liu, and Y. Zhang, "Fairness in recommendation: Foundations, methods, and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 5, pp. 1–48, 2023.

[32] L. E. Celis, D. Straszak, and N. K. Vishnoi, "Ranking with fairness constraints," *arXiv preprint arXiv:1704.06840*, 2017.

[33] D. Serbos, S. Qi, N. Mamoulis, E. Pitoura, and P. Tsaparas, "Fairness in package-to-group recommendations," in *Proceedings of the 26th international conference on world wide web*, pp. 371–379, 2017.

[34] H. Abdollahpouri and R. Burke, "Multi-stakeholder recommendation and its connection to multi-sided fairness," *arXiv preprint arXiv:1907.13158*, 2019.

[35] R. Burke, "Multisided fairness for recommendation," *arXiv preprint arXiv:1707.00093*, 2017.

[36] G. K. Patro, A. Chakraborty, N. Ganguly, and K. Gummadi, "Incremental fairness in two-sided market platforms: On smoothly updating recommendations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 181–188, 2020.

[37] K. Yang and J. Stoyanovich, "Measuring fairness in ranked outputs," in *Proceedings of the 29th international conference on scientific and statistical database management*, pp. 1–6, 2017.

[38] E. Streviniotis, A. Georgara, F. Bistaffa, and G. Chalkiadakis, "Fairplay: A multi-sided fair dynamic pricing policy for hotels," 2024.

[39] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates, "Fa* ir: A fair top-k ranking algorithm," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1569–1578, 2017.

[40] T. Carter, "An introduction to information theory and entropy," *Complex systems summer school, Santa Fe*, 2007.

[41] T. M. Cover, J. A. Thomas, *et al.*, "Entropy, relative entropy and mutual information," *Elements of information theory*, vol. 2, no. 1, pp. 12–13, 1991.

[42] C. C. Aggarwal *et al.*, *Recommender systems*, vol. 1. Springer, 2016.