Algorithmic Bias in Graph-Based Recommender Systems

Francesco Fabbri

TESI DOCTORAL UPF / year 2022

Directors de la tesi
Dr. Francesco Bonchi & Dr. Carlos Castillo

Department of Information and Communication Technologies

To my mum, my dad and my sister.

Acknowledgment

This thesis is the outcome of a 4-years-long journey and along the way, many are the ones I should thank to have collaborated with me, supported me and contributed to reach this moment.
First, I want to thank my advisors, Francesco Bonchi and Carlos Castillo. I will always be grateful for their patience, guidance, for what they taught me and for the way we got along. They shared with me so much experience, knowledge and wisdom in these 4 years, that I couldn't feel luckier.
Then thanks to my colleagues at Eurecat, in particular to Ludovico, Cristian, Pablo, David and Julia for sharing with me all those meaningful conversations and valuable pieces of guidance along the way.
Being part of DTIC Department at UPF, I had the chance to meet such great individuals and grow enriching friendships. A particular mention to Lydia Garcia, who had the patience to solve any paperwork or administrative issue I would submit to her.
I had the pleasure to share this journey along with many Ph.D. students and researchers at UPF. I can not forget my very own beloved "Queens & Boomers", which is not only a group of people doubtfully good at playing beach volley, but primarily a great group of friends, which have been part of this journey since almost day 0. In particular, I want to thank Silvia, my beloved "deskmate", whom I was lucky enough to share the Ph.D. journey, through a rollercoaster of emotions. I also would like to mention Marina, who became so close to me in such a short period of time and with whom I collected so many nice memories already.
I am also grateful for the experience and moments in Helsinki. In particular, I want to thank Michael for being an incredible host at the University of Helsinki. Furthermore, I want to thank all the friends I made there, that made me living an incredible 5-months period.
Thanks to Giacomo, a friend that been sharing this journey with from my very first days in Barcelona. We have shared countless memories together that would request too long to be written here.
I was lucky enough to meet along the way such valuable and enriching people, whom, even if sometimes only for a portion of the journey, shared

with me their time along with intense and nice moments. Among them, thanks to Silvia, Eleonora, Camilla, Helena, Dani, Ibrahim, Maria Luisa and Elena.
A very special thanks to Lea, that in a very short amount of time became pivotal and crucial supporting me in this last part of the journey. I can't say enough how grateful and lucky I feel to have her on my side.
Finally, thanks to my family, my mum Antonella, my dad Salvatore and my sister Giulia. The way I worked and how I ended up this journey is mainly because of them. Their love, their guidance and their support is hard to quantify and even explain. This thesis wouldn't have been possible without them, thanks for everything.

Abstract

Recommender Systems represent a key instrument to convey consumption of contents available on the Web. They enhance the engagement among the users and the online platforms through algorithmic personalization. Injecting non-natural interactions consequently cannot have only beneficial effects. Indeed, amplifying and exaggerating human behaviors leads to either the spread of extreme point of views (e.g. polarized or controversial opinions) or the discrimination or mistreatment of a specific group of individuals. In this thesis, we pose the attention on the importance of auditing and mitigating the "algorithmic bias" generated by a recommendation system, emphasizing its role on the networked interactions of users and contents. Through empirical evidences we highlight how the social graph, presenting biased network topology, when used as input, can impact the algorithmic recommendations. This analysis allows to add a perspective on the long-term impact of algorithmic suggestions, leading to design a simulation model able to explain the "feedback-loop" generated on social networks. Auditing the algorithmic bias facilitates the design of strategies able to mitigate algorithmic risks in recommendation, such as radicalization and unfairness. The results found in this thesis raise critical observations about the impact of recommendation algorithms, and hints of the need to design systems able to mitigate biases embedded in data and algorithms, considering both short and long-term perspectives.

Resumen

Los sistemas de recomendación representan un instrumento clave para vehicular el consumo de contenidos disponibles en la Web. Mejoran el vínculo entre los usuarios y las plataformas en líne a través de la personalización algorítmica. En consecuencia, la inyección de interacciones no naturales no tiene sólo efectos positivos. La amplificación y exageración de los comportamientos humanos conduce a la difusión de puntos de vista extremos (por ejemplo, opiniones polarizadas o controvertidas) y a la discriminación o el maltrato de un grupo específico de individuos. En esta tesis, se pone la atención en la importancia de auditar y mitigar el ”sesgo algorítmico” generado por un sistema de recomendación, enfatizando su función en las interacciones en redes de usuarios y de contenidos. A través de evidencias empíricas evidenciamos cómo el grafo social, que presenta una topología de red sesgada, puede impactar en las recomendaciones algorítmicas, cuando se utiliza como input. Este análisis permite añadir una perspectiva sobre el impacto a largo plazo de las sugerencias algorítmicas, llevando a diseñar un modelo de simulación que permite de explicar el “feedback-loop” por las mismas en las redes sociales. La comprobación del sesgo algorítmico facilita el diseño de estrategias capaces de mitigar los riesgos algorítmicos en la recomendación, como la radicalización y la injusticia. Los resultados obtenidos plantean observaciones críticas sobre el impacto de los algoritmos de recomendación, e insinúan la necesidad de diseñar sistemas capaces de mitigar los sesgos incorporados a los datos y a los algoritmos, teniendo en cuenta tanto las perspectivas a corto como a largo plazo.

Resum

Els sistemes de recomanació representen un instrument clau per vehicular el consum de continguts disponibles a la web. Milloren el compromís entre els usuaris i les plataformes en línia mitjançant la personalització algorítmica. Per tant, la injecció d'interaccions no naturals no només té efectes positius. L'amplificació i l'exageració dels comportaments humans condueix a la difusió de punts de vista extrems (per exemple, opinions polaritzades o controvertides) o a la discriminació o el maltractament d'un grup específic d'individus. En aquesta tesi, es posa l'atenció en la importància d'auditar i mitigar el "biaix algorítmic" generat per un sistema de recomanació, emfatitzant-ne la funció en les interaccions en xarxes d'usuaris i continguts. A través d'evidències empíriques evidenciem com el graf social, que presenta una topologia de xarxa esbiaixada, pot impactar en les recomanacions algorítmiques quan s'utilitza com a input. Aquesta anàlisi permet afegir una perspectiva sobre l'impacte a llarg termini dels suggeriments algorísmics, portant a dissenyar un model de simulació que permet explicar el "feedback-loop" generat per aquestes a les xarxes socials. Aquesta anàlisi va facilitar el disseny d'estratègies capaces de mitigar els riscos algorítmics en la recomanació, com ara la radicalització i la injustícia. Els nostres resultats plantegen observacions crítiques sobre l'impacte dels algorismes de recomanació, i insinuen la necessitat de dissenyar sistemes capaços de mitigar els biaixos incorporats a les dades i als algoritmes, considerant tant les perspectives a curt com a llarg termini.

Sommario

I sistemi di raccomandazione rappresentano uno strumento fondamentale per veicolare il consumo dei contenuti disponibili sul Web. Questi sistemi migliorano il coinvolgimento degli utenti e delle piattaforme online attraverso la personalizzazione tramite algoritmi di intelligenza artificiale. Iniettare interazioni non naturali, tuttavia, non ha solo effetti benefici. Infatti, amplificare ed esagerare i comportamenti umani porta alla diffusione di punti di vista estremi (ad esempio, opinioni polarizzate o controverse) e alla discriminazione o al maltrattamento di un gruppo specifico di individui. In questa tesi, poniamo l'attenzione sull'importanza di verificare e mitigare il "bias algoritmico" generato da un sistema di raccomandazione, enfatizzando il suo ruolo nelle interazioni in reti di utenti e di contenuti. Attraverso evidenze empiriche si evidenzia come la rete sociale distorta, se usata come input, possa avere un impatto sulle raccomandazioni. Questa analisi permette di aggiungere una prospettiva sull'impatto a lungo termine dei suggerimenti automatici, portando a progettare un modello di simulazione in grado di spiegare il "feedback-loop" generato sulle reti sociali. Inoltre, la valutazione dei bias algoritmici facilita la progettazione di strategie in grado di mitigare effetti dannosi nelle raccomandazioni, come la radicalizzazione e le disparità. I risultati di questa tesi sollevano osservazioni critiche sull'impatto dei sistemi di raccomandazione e accennano alla necessità di progettare soluzioni in grado di mitigare i bias incorporati nei dati e negli algoritmi, considerando prospettive sia a breve che a lungo termine.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Context

Information Technologies have a crucial influence on the way people get access to online content. Spotify can count 460M of monthly active users, while Twitter 216M, TikTok reaches even 1B. In these scenarios, it becomes challenging for customers to find the right product to consume or video to watch due to the growing volume, variety, and growth speed of items online. Artificial Intelligence (AI) algorithms became a pivotal instrument to solve these kind of tasks, by understanding user's needs, and supporting her for the next decision to take (e.g. what to consume or what to buy) while browsing through Online Social Platforms (OSPs). The introduction of these intelligent systems into the OSPs' personalisation represents indeed a key factor to guarantee increases in user engagement and long-term retention [RRS11, AMA+20, QCJ18, ZHW+19]. Specifically, Information Retrieval (IR) and Recommender Systems (RS) algorithms are the engines of these features.

IR and RS models can infer user behavior, grasping similarities with other individuals using the same OSPs, to eventually filter out and select items that a user might like. Based on the personalization task, the following categorization of OSPs can be proposed: the platform can be

a "content sharing website" like Spotify or YouTube, in which the crucial task is recommending the next song, video (e.g. "What To Watch Next") or playlist (e.g. "Weekly Discovery") to keep the users interacting with the online service. It can also be a *"social networking website"*, like Facebook (e.g."People you may know") or Twitter (e.g. "Who To Follow") which, suggestions of new friend to connect with or to follow, are a great boost for the growth of the user network, expanding the list of user friends and increasing the chances of connecting with different individuals. In both scenarios, a recommendation algorithm is defined as successful if able to personalize the most the user experience, evaluated in terms of metrics such as accuracy or click-through rate (CTR) [WWY15, KJJ18]. This point of view results to be narrow, and loses the perspective on issues that also affect sociotechnical aspects involved in the personalization. Only recently, it has been possible to detect how RS, while optimizing for accuracy, can embed and exaggerate human-biases included in either the algorithm design or training data. It becomes crucial then, for both practitioners and scholars, to consider the sociotechnical implications due to the injection of different types of stereotypes or beliefs into the life cycle of a AI solution based, like recommender systems [SG21, MSWVW20].

Several are the examples of harmful actions triggered by an algorithmic recommendation or ranking. For example, Amazon has been developing a recruiting engine to filter applicants resumes with the aim of automating the search for top talent. It was only just before deploying that the company realized its automated system was not rating candidates in a gender-neutral way. Indeed, women were discriminated because the data used to training the model was a collection of the last 10 years of received resumes, which were coming mostly from male candidates, reflecting a male dominance across the tech industry [Das18]. The selection algorithm was under-exposing women, which represents a minority in the tech industry, reinforcing the stereotype already existing in society.

Unfairness against minorities is not the only harm experienced through recommendation algorithms. Indeed, if not moderated, the algorithm outcome can represent a medium for controversial or even toxic content

[AG05, SPGK19]. For example, YouTube recommendation algorithm has shown to be affected by the radicalized interactions happening on the platform. In particular, it has been observed how communities presenting huge volumes of interactions upon controversial topics may lead the recommendation algorithm to be biased, driving the neutral or non-polarized users into the "rabbit hole" of controversial content [LZ20]. Among many, these examples highlight how "algorithmic bias" represents an emergency that needs to be addressed. There is an urgency to characterize, but also design, new solutions able to detect and mitigate harmful and discriminative actions stimulated by the interactions between the user and the recommendation output. However, introducing new bias-aware algorithms remains a challenging task because they have to be built up on existing user-centric methods, thought to optimize quite different metrics, like accuracy or relevance [Kle18, WWB+21, KMR17a]. Moreover, aspects related to time and space are transversal in these settings and cannot be underestimated into the design of large-scale RS. Static view of the problem formulation is not sufficient, since it lacks of perspective on the recurrent user-item interactions. Indeed, OSPs are characterized by continuous interactions between user and items, which open to new algorithmic challenges, faced when trying to either study inequalities incorporated in the output, but also when trying to reduce any harmful effect on the long term. It becomes crucial then to target long-term goals coping with fairness, transparency and accountability, and designing a system that dynamically addresses sociotechnical issues in which data may be received sequentially (e.g. streaming data).

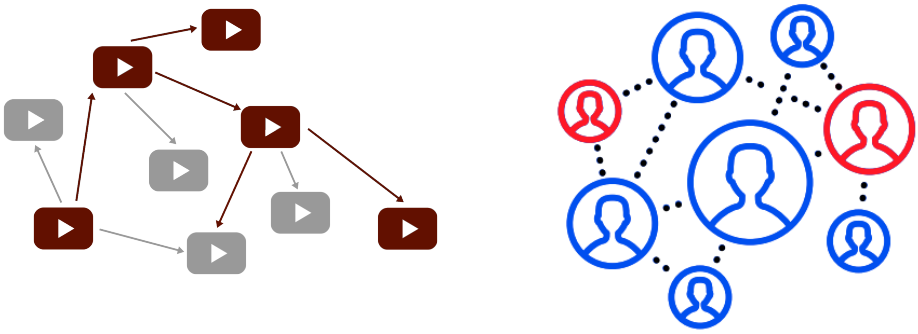
1.2 Research Goals

The scope of this thesis lies at the intersection of Recommender Systems and Computational Social Science. The main goal of this manuscript is to characterize and mitigate the impact of the human biases involved into the design and training of recommendation algorithms. Specifically, we analyze the impact of human beliefs and stereotypes embedded into the RS,

characterizing the phenomena impacting the algorithmic output and then proposing strategies to incorporate bias-aware metrics into either during (in-processing) or after (post-processing) the training. To do so, we follow a multiple stakeholders' perspective for the analysis of RS [AAB+19], in which both consumers (e.g. users) and providers (e.g. whom producing content) are considered. To leverage this dual perspective, we use network data structures, which allow to smoothly perform analysis between users and recommendation algorithms. Through the manuscript the network representation is indeed the recurrent element binding our different contributions.

In the case of "social networking website", users can be represented as social graph, in which two nodes (u, v) are connected because of their online friendship, and receive suggestions through recommendation algorithms.

Also, "content sharing website", in which sequential interactions happen between users and suggested content, allows to build a network of content, displaying the dependencies between recommended items. In this case, the algorithm output can be designed as a video-to-video network presenting connections where an edge (u, v) exists if it is possible to jump from content u to content v through a recommendation.



(a) Network of Content (b) Network of Users

Figure 1.1: Network Representation of User and Video interactions

In this manuscript network representations are key components to characterize and mitigate the different type of harmful algorithmic biases stimulated by the RS. For the "social networking website", we monitor homophily into the social graph, showing how it plays a key role impacting the level of inequality generated by the output exposure. Thanks to the insights extracted by studying the network of users, we shed the light on the direct impact of human bias over the algorithm, expressed in terms of disparate exposure among the users. To extend this analysis and study the indirect impact of the algorithmic effect over the network of users, we introduce a new simulation model, able to give insights on the long-term scenario, in which multiple interactions between user and algorithm are generated. We investigate the "feedback-loop" generated by the RS, through a simulation model, capturing the interplay between network of users, recommendation algorithm, and user feedback. Characterizing the sequential interactions between user and algorithms through a simulation framework allows us to design new strategies able to mitigate similar kinds of structural bias in "content sharing website". Indeed, we devise new algorithms to reduce structural bias in "what-to-watch-next" recommender systems, modifying the sequential interactions between user and content. Specifically, we study how a radicalized group of users, included in the training of the RS, can eventually have an impact on the sequential browsing of the output, generating a "rabbit-hole" of radicalized content in which a user may be stuck in. Mitigating the algorithmic bias in an online fashion opens to design methods able to deal with data coming in batches and streaming. For this reason, we investigate strategies to maintain bias-aware constraints over the output, able to deal with input streaming data. Specifically, the challenge in this work is to design algorithms that include bias-aware constraints able to deal with the "price of fairness", which represents the impossibility to generate an outcome that maximizes fairness and accuracy at the same time. This constraint encouraged us to design new techniques able to minimize the bias with a relatively small loss in relevance. The proposed solutions are tested in settings fitting both "social networking website" and "content sharing website" scenarios.

1.3 Thesis Contribution

In this thesis we study and characterize Algorithmic Bias in Recommender Systems, introducing metrics to quantify the biases included in the recommendation output and proposing strategies to mitigate those biases. The research outcome is divided in two parts: (1) the first part focuses on presenting the background and the recent state of the art; (2) the second part focuses on characterizing algorithmic bias in PRS; (3) the third part focuses on proposing strategies to mitigate algorithmic biases in W2W and in presence of streaming data.

Part 1: Background and State of the Art

The chapters of this part are needed to present the current state of the art connected to our work and introduce related definitions.

Chapter 2. In this chapter we present the background on recommender systems, algorithmic polarization and algorithmic fairness.

Chapter 3. In this chapter we review the state of the art related to characterisation and mitigation of harmful algorithmic biases in OSPs, such as algorithmic polarization and discrimination.

Part 2: Bias in People Recommender Systems

In the 2nd Part we first analyze the impact of user homophily in PRS, after one round of recommendations. Then, we propose a simulation model to capture long-term effects of homophily over the same type of algorithms.

Chapter 4. We investigate the exposure of minorities in people recommender systems in social networks. Specifically, we consider a bi-populated social network, i.e., a graph where the nodes belong to two different groups (majority and minority) and, by applying state-of-the-art

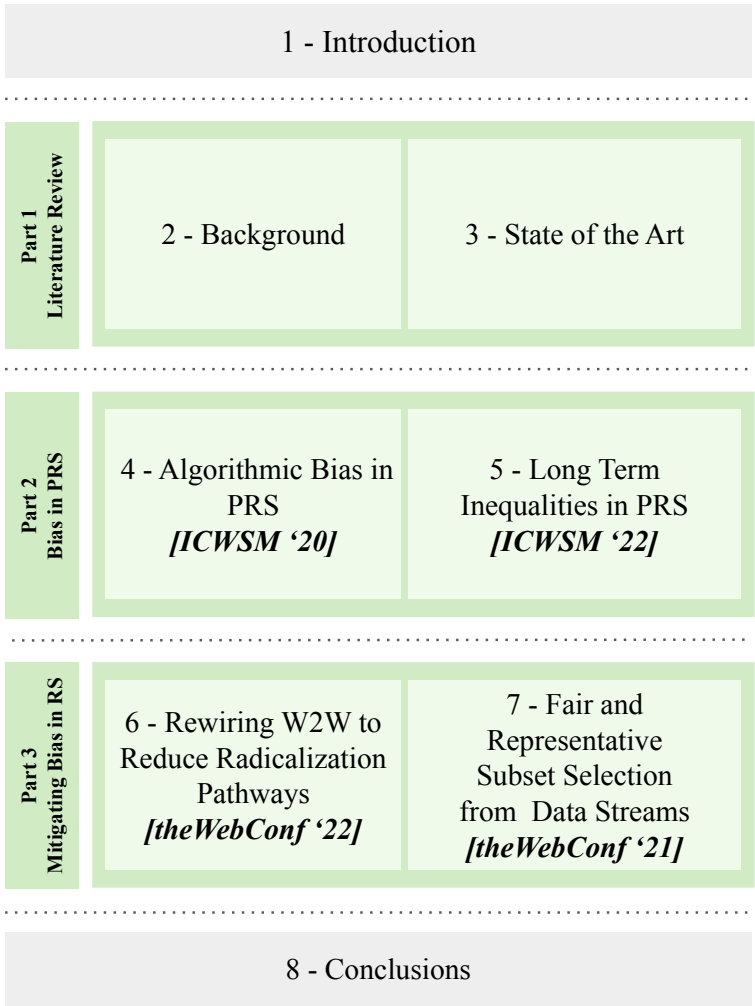


Figure 1.2: Thesis Structure

people recommenders, we analyze how disparate exposure can be amplified or mitigated by different levels of homophily within each subgroup. We start our analysis on real-world social graphs, where the two subgroups are defined by sensitive demographic attributes such as gender or age. Our findings suggest that the way and the extent to which people recommenders can produce disparate exposure on the two subgroups, might depend in large part on the level of homophily within the subgroups. To verify these findings, we move our analysis to synthetic datasets, where we can control characteristics of the input social graph, such as the size of the minority and the level of homophily. Our results show that homophily plays a key role in promoting or reducing exposure for different subgroups under various combinations of dataset characteristics and recommendation algorithms. The work described in this chapter has been published in:

[FBBC20] *Fabbri, F., Bonchi, F., Boratto, L., & Castillo, C. (2020, May). The effect of homophily on disparate visibility of minorities in people recommender systems. In Proceedings of the International AAAI Conference on Web and Social Media (Vol. 14, pp. 165-175).*

Chapter 5. In this chapter we investigate PRS effects, introducing a model to simulate the feedback loop created by multiple rounds of interactions between users and a link recommender in a social network. This allows us to study the long-term consequences of those particular recommendation algorithms. Our model is equipped with several parameters to control (i) the level of homophily in the network, (ii) the relative size of the groups, (iii) the choice among several state-of-the-art link recommenders, and (iv) the choice among three different user behavior models, that decide which recommendations are accepted or rejected. Our extensive experimentation with the proposed model shows that a minority group, if homophilic enough, can get a disproportionate advantage in exposure from all link recommenders. Instead, when it is heterophilic, it gets under-exposed. Moreover, while the homophily level of the minority affects the speed of the growth of the disparate exposure, the relative

size of the minority affects the magnitude of the effect. Finally, link recommenders strengthen exposure inequalities at the individual level, exacerbating the "rich-get-richer" effect: this happens for both the minority and the majority class and independently of their level of homophily. The model and the results described in this chapter are presented in:

[FCBC21] *Fabbri, F., Croci, M. L., Bonchi, F., & Castillo, C. (2021). Exposure Inequality in People Recommender Systems: The Long-Term Effects. In Proceedings of the International AAAI Conference on Web and Social Media (ICWSM '22).*

Part 3: Mitigating Bias in Recommender Systems

In the 3rd part of the manuscript we propose strategies to mitigate radicalisation pathways generated through the W2W and solutions to reduce unfairness in scenarios dealing with streaming data.

Chapter 6. Recommender systems typically suggest to users content similar to what they consumed in the past. If a user happens to be exposed to strongly polarized content, she might subsequently receive recommendations which may steer her towards more and more radicalized content, eventually being trapped in what we call a "*radicalization pathway*". In this chapter, we study the problem of mitigating radicalization pathways using a graph-based approach. Specifically, we model the set of recommendations of a "*what-to-watch-next*" recommender as a d-regular directed graph where nodes correspond to content items, links to recommendations, and paths to possible user sessions. We measure the "*segregation*" score of a node representing radicalized content as the expected length of a random walk from that node to any node representing non-radicalized content. High segregation scores are associated to larger chances to get users trapped in radicalization pathways. Hence, we define the problem of reducing the prevalence of radicalization pathways by selecting a small number of edges to "*rewire*", so to minimize the maximum of segregation scores among all radicalized nodes, while maintaining the

relevance of the recommendations. We prove that the problem of finding the optimal set of recommendations to rewire is NP-hard and NP-hard to approximate within any factor. Therefore, we turn our attention to heuristics, and propose an efficient yet effective greedy algorithm based on the absorbing random walk theory. Our experiments on real-world datasets in the context of video and news recommendations confirm the effectiveness of our proposal. The algorithms and results presented in this chapter have been published in:

[FWB+22] *Fabbri, F., Wang, Y., Bonchi, F., Castillo, C., & Mathioudakis, M. (2022, April). Rewiring What-to-Watch-Next Recommendations to Reduce Radicalization Pathways. In Proceedings of the ACM Web Conference 2022 (pp. 2719-2728).*

Chapter 7. We study the problem of extracting a small subset of representative items from a large data stream. In many data mining and machine learning applications, such as social network analysis and recommender systems, this problem is formulated as maximizing a monotone submodular function subject to a cardinality constraint k. In this work, we consider a setting where data items in the stream belong to one of several disjoint groups and investigate the optimization problem with an additional *fairness constraint* that limits selection to a given number of items from each group. We propose efficient algorithms for this fairness-aware variant of the streaming submodular maximization problem. In particular, we first give a $(\frac{1}{2} - \varepsilon)$-approximation algorithm that requires $O(\frac{1}{\varepsilon} \cdot \log \frac{k}{\varepsilon})$ passes over the stream for any constant $\varepsilon > 0$. In addition, we provide a single-pass streaming algorithm that has the same $(\frac{1}{2} - \varepsilon)$ approximation ratio when an unlimited buffer size and post-processing time are permitted, and discuss how to adapt it to practical settings with bounded buffer sizes. Finally, we demonstrate the efficiency and effectiveness of our proposed algorithms on two real-world applications, namely *maximum coverage on large graphs* and *personalized recommendation*. The results and the algorithmic solutions presented in this chapter have been published in:

[WFM21] *Wang, Y., Fabbri, F., & Mathioudakis, M. (2021, April). Fair*

and representative subset selection from data streams. In Proceedings of the Web Conference 2021 (pp. 1340-1350).

1.4 Additional Output

During my studies I had the opportunity to contribute also to scientific articles, which are not part of this manuscript.

Session-Based RS. Session-based recommender systems consider the evolution of user preferences in browsing sessions. Existing studies suggest as next item the one that keeps the user engaged as long as possible. This point of view does not account for the providers' perspective. In this work, we highlight side effects over the providers caused by state-of-the-art models. We focus on the music domain and study how artists exposure in the recommendation lists is affected by the input data structure, where different session lengths are explored. We consider four session-based systems on three types of datasets, with long, short, and mixed playlist length. We provide measures to characterize disparate treatment between the artists, through a systematic analysis by comparing (i) the exposure received by an artist in the recommendations and (ii) their input representation in the data. Results show that artists for which we can observe a lot of interactions, but offering less items, are mistreated in terms of exposure. Moreover, we show how input data structure may impact the algorithms' effectiveness, possibly due to preference-shift phenomena.

[AFBS21] *Ariza, A., Fabbri, F., Boratto, L., & Salamo, M. (2021, March). From the Beatles to Billie Eilish: Connecting Provider Representativeness and Exposure in Session-Based Recommender Systems. In European Conference on Information Retrieval (pp. 201-208). Springer, Cham.*

Algorithmic Bias in double-sided markets. Machine Learning (ML) techniques have been increasingly adopted by the real estate market in the last few years. Applications include, among many others, predicting

the market value of a property or an area, advanced systems for managing marketing and ads campaigns, and recommendation systems based on user preferences. While these techniques can provide important benefits to the business owners and the users of the platforms, algorithmic biases can result in inequalities and loss of opportunities for groups of people who are already disadvantaged in their access to housing. In this work, we present a comprehensive and independent algorithmic evaluation of a recommender system for the real estate market, designed specifically for finding shared apartments in metropolitan areas. We were granted full access to the internals of the platform, including details on algorithms and usage data during a period of 2 years. We analyze the performance of the various algorithms which are deployed for the recommender system and asses their effect across different population groups. Our analysis reveals that introducing a recommender system algorithm facilitates finding an appropriate tenant or a desirable room to rent, but at the same time, it strengthen performance inequalities between groups, further reducing opportunities of finding a rental for certain minorities.

[SFC+21] *Solans, D., Fabbri, F., Calsamiglia, C., Castillo, C., & Bonchi, F. (2021, July). Comparing Equity and Effectiveness of Different Algorithms in an Application for the Room Rental Market. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (pp. 978-988).*

RS preserving users' privacy. Rigorous data protection regulations such as EU/UK General Data Protection Regulation (GDPR) has made the demand for technology that protects sensitive user data ever more important. Federated Learning (FL) has become a pivotal formulation for distributing the computation of machine learning models to improve online services personalisation and at the same time preserving the privacy of the users. In contrast to traditional centralised learning approaches where private user data is transmitted to a central server, in an FL setting computations are distributed among multiple devices which first train the model locally and then share their local updates with a global server, coordinating the decentralized computation between the devices.

In this decentralised setting, ensuring that models can be trained fast and accurately is a pivotal task. This is especially true for recommendation algorithms, which, if slow to train, may provide poor quality recommendations and result in a loss of user engagement. In this work, we propose FedFNN, a novel algorithm which speeds up the training of models in decentralised environments.

In the FL setting, only a subset of users are sampled for training at each epoch. FedFNN uses supervised learning to predict weight updates of unsampled clients, training the model on the local updates of the sampled clients. We show the effectiveness of our approach on both real and synthetic data. In particular, we show that: (i) FedFNN is on average 5x faster than the current state-of-the-art and can provide a better level of accuracy with the same number of iterations; (ii) through synthetically generated data we observe that the number of client clusters does not affect the performance of FedFNN; (iii) FedFNN is more robust to the problem of poor client availability and in such scenarios converges faster than the current state-of-the-art.

Fabbri F., Lui X. McKenzie1 J., Twardowski B. &Kurniawan Wijaya T., FedFNN: Faster Training Convergence Through Update Predictions in Federated Recommender Systems, 2022, under review

Part I

Background and State of the Art

2 Background

2.1 Recommender Systems

Recommender systems are a core functionality in online social platform, which the task is to personalize user experience, trying to predict next user responses or interactions to alternatives mostly based on people's behaviour, taste or feedback. A crucial distinction of RS can be done by type of interactions. Indeed, the interactions can either be: (i) *item-to-user*, which are between users and contents, like in the case of streaming content services (e.g. Netflix, Spotify, YouTube); (ii) *user-to-user*, in which the users interact between themselves in a social network (e.g. Twitter, LinkedIn, Facebook). The first one can be while the second one in "social networking website". In the case of user-to-user interactions on "content sharing website", *"People Recommender Systems"* (PRS) [GP16, LFS17b] are one of the most popular algorithms, in which the recommendations are based on the network topology of the social network. In the case of item-to-user, *"What To Watch next"* (W2W) recommenders are one of the popular solutions, included in systems where the suggested item is usually consumed immediately after the current one [ZHW+19, SJC+20]. Fig. 2.1 shows an example of W2W recommendations, popular in the case of video streaming platforms.

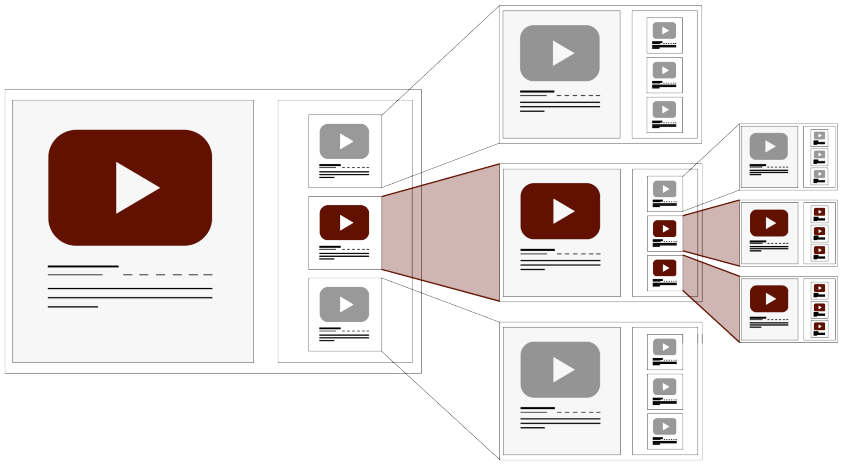
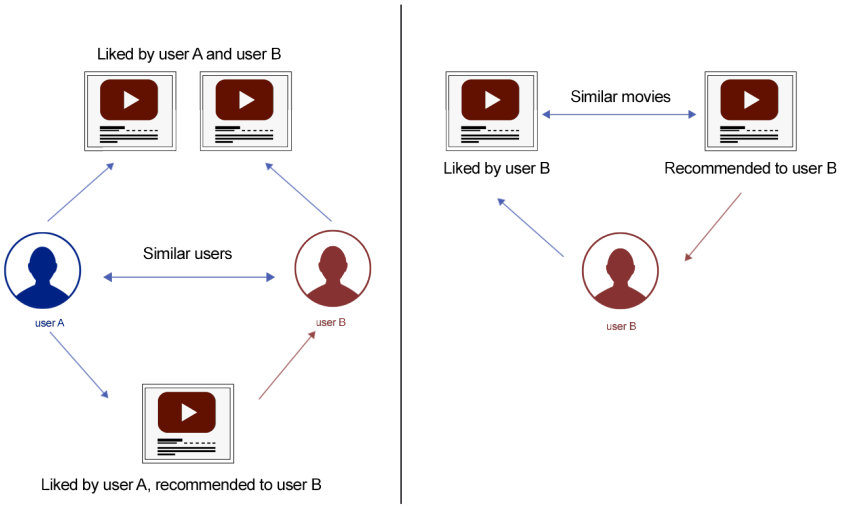


Figure 2.1: What-To-Watch-Next Recommender System in an online video streaming platform.

Both types of interactions, and consequently both recommendation frameworks PRS and W2W, are strongly dependent on the type of features included in the input data, which will affect the choice of the model design. Indeed, depending on the features, we can distinguish between: (i) *collaborative filtering* models, which (**CF**) try to predict future user behavior using interactions between user and recommended contents; (ii) *content-based* models (**CB**), which predict the user behavior, based on similarities between content. In the Figure 2.2 we present an example to better explain these differences. In the left figure, following the CF approach, a new content is recommended to user B, because of similar tastes with user A (e.g. user A and B have interacted with same videos). While in the right side, based on CB, a new content is recommended to user B only because of similarities between content.

Collaborative Filtering. In CF the attention is on the interactions' matrix $\mathbf{M}$, which has on the rows the set of N users $\mathbf{U} = \{u_1, ..., u_N\}$ and on the columns the set of M contents $\mathbf{I} = \{i_1, ..., i_M\}$. From all the



(a) Collaborative Filtering (b) Content Based

Figure 2.2: Collaborative Filtering (CF) and Content Based (CB) Approaches for Recommender Systems.

possible interactions $\mathbf{M}$, it is possible to extract the subset of the ones observed $\mathbf{M}^{obs} \subseteq \mathbf{M}$ where $m_{ui} \neq 0$ if the user u has interacted with the item i. The final task is to predict for each user, the missing interactions $\hat{\mathbf{M}} = \mathbf{M} \setminus \mathbf{M}^{obs}$. Also, interactions of the users can be either binary, also called implicit feedback or categorical (e.g. ratings from 1 to 5), also called explicit feedback. Here we focus on implicit feedback. Generally, CF algorithms can be applied to both W2W and PRS, and the most popular one is based on matrix factorization, which the final goal is predict the missing user-item interactions, through the generation of low-dimensional embeddings to represent both users and items.

We show first the of W2W (user-ot-item), indicating with $\mathbf{v}_u$ and $\mathbf{v}_i$ respectively the user and content embeddings, the objective is to find a set of low-dimensional vectors able to minize the following objective function:

$$\min_{m_{ui} \in \mathbf{M}} \sum_{u,i} (m_{ui} - \mathbf{v}_u^T \mathbf{v}_i)^2$$

The inner product between the user and item embeddings represents the estimated relevance of the item i for the user u. Starting from the formula above is possible to derive different approaches, which differs based on the applications and the data characteristics. Among many, Alternating Least Squares (ALS) that works with implicit feedback, represents the standard in CF [HKV08], which the objective is represented by:

$$\min_{m_{ui} \in \mathbf{M}} \sum_{u,i} (m_{ui} - \mathbf{v}_u^T \mathbf{v}_i)^2 + \lambda(||\mathbf{v}_u||^2 + ||\mathbf{v}_i||^2)$$

Adapting this algorithm to PRS can be done just through the definition of the input matrix. Indeed, the matrix of interactions is given by the adjacency matrix $\mathbf{A}$ of a social graph $\mathcal{G} = (V, E)$, where $\mathbf{V}$ represents the set of users in the graph and E the set of edges, where $a_{ui} = 1$ if an edge between user u and user i exists [SC18]. The task in this case is to predict the next link in the graph and the CF algorithm can still be used to generate low dimensional vectors of users, which are now on both rows and columns. For the item-to-user models the similarity between users is captured in order to find out similar history in contents' consumption, while in the user-to-user models the similarity is captured in order to find similar patterns in friendships in the social graph.

2.2 Algorithmic Harms in Recommender Systems

In the literature, RS have be usually designed and thought to be *"user-centric"*, with the objective to maximize user personalization. However,

this strategy has been proven to fall short in evaluating the influence of the RS output upon other stakeholders involved in the platforms, like content providers and platforms' owner, which are actively involved in the recommendation process. In a recent survey, it has been introduced the *multi-stakeholder* framework in recommender systems, which opens to design recommenders able to include not only the consumers, but also the providers perspective into the design of the algorithms [AAB+19]. As stated in the survey:

Definition 1. *(Stakeholder). A recommendation stakeholder is any group or individual that can affect, or is affected by, the delivery of recommendations to users.*

In this thesis we find evidences of issues affecting different stakeholders included in the pipeline by different user-centric approaches. We also propose strategies to mitigate those. Through this perspective it is possible to highlight socio-technical issues affecting one or many stakeholders. Indeed, having this user-centric approach which solely focuses on optimizing accuracy, different forms of human-bias can be included and spread through a personalization algorithm [EBD19].

In this manuscript we embrace the multistakeholder perspective, analyzing how different forms of bias are distributed and perpetuated within the algorithm output. The definition of *bias* proposed by Baeza-Yates is the one fitting the most our setting [BY18]:

Definition 2. *(Bias). From a statistical point of view, bias is a systemic deviation caused by an inaccurate estimation or sampling process. As a result, the distribution of a variable could be biased with respect to the original, possibly unknown, distribution.*

In agreement with this definition of bias, a recent survey refined the categorization of different potential sources and forms of biases included in a system implementing Machine Learning solutions [MMS+21]. In particular, a categorization of three different groups of bias is proposed:

(i) *from Data to Algorithm*, in which the biased distribution of input data leads to a biased outcome;

(ii) *from Algorithm to User*, in which the system design is biased by human beliefs and affect the output;

(iii) *from User to Data*, which is present when a biased user behavior affects the data collection.

In this thesis we cope with biases at the intersection of the first two categories, also called "Algorithmic Bias". This phenomenon is generated by a combination of effects of training data and algorithmic design.

2.2.1 Algorithmic Polarization

Online Social Platforms (OSPs) are well-known to be characterized by power-law distributions [BA99] which naturally lead to differences in nodes' degree distribution. Those inequalities in networks derives from social structures that can create groups and patterns of inequality mediated by information access (e.g. links distribution) [BLM14].

A well-known phenomenon that quantifies those inequalities is called "rich-get-richer" effect, which shows how nodes degree grows sub-linearly, i.e. nodes with higher degree grows faster than smaller ones. This growth benefits only a subset of nodes that will get most of the connections, leaving the vast majority with few numbers of connections [EK10].

Node degree represents also a social capital which, if allocated disproportionally, can lead to discrimination in recommendation [VSFC21, Bur00].

Indeed, ML algorithms play a crucial role in the distribution of this social capital, since they can modify and augment the original network topology, potentially stimulating the differences in attention among nodes, affecting the level of information access among users. Effects characterizing segregated network are reconnected to Social Network Polarization, in which users tend to reinforce their own beliefs and point of views, distributing their connections and interactions far from different opinions [GW17]. Several evidences show how algorithmic curation results not only in boosting Online Polarization, but the creation of *"filter bubbles"* [Par11]:

Definition 3. *(Filter Bubble). Filter Bubble results in a type of tunnel vision, effectively isolating people into their own cultural or ideological bubbles.*

2.2.2 Algorithmic Fairness

Algorithmic Fairness research focuses on characterizing and mitigating possible human biases that algorithms may exaggerate either inherited from the data or derived by its design [BS14, Cho17]. A universal fairness definition cannot be delineated , since even if rigorous statistically defined, it happens to be unfeasible to apply in all the possible contexts [Nar18]. Preferring one among the others is highly dependent by the context and recent work also proves the impossibility of meeting at the same time more than one definition of Algorithmic Fairness in a supervised learning setting [KMR17b].

According with the extensive literature produced recently in Fairness in Machine Learning, especially in supervised learning, we can distinguish three statistical non-discrimination criteria [BHN19, CR18]:

Definition 4. *(Independence). This Criterion requires the sensitive characteristic to be statistically independent of the score. This definition implicitly assumes there are no intrinsic differences between different protected group features, which represents a big limitation, since it never holds in reality.*

Definition 5. *(Separation). This criterion overcomes the first one, demanding independence within each stratum of the population defined by target variable, not only globally without taking into account the outcome of the target. This criterion is defined through conditional probably over the requirement fairness over the target.*

Definition 6. *(Sufficiency). This criterion assumes the condition for which the score incorporates the sensitive characteristic useful to predict the target, and conditions the output to be independent from that.*

Those criteria are partially limited by the possibility to observe the relationship existing between sensitive outcome, target and output variables. As an alternative of these criteria, the definition of individual fairness has been introduced, which aims for the following: *"similar individuals should be treated similarly"* [DHP+12]. The challenge in this case, is to define the right distance metrics in order to define similarity between users (highly dependent on the context).

The seminal work developed addressing Algorithmic Discrimination in Machine Learning have stimulated the growth of this new research directions also in the area of IR and RS [EBD19]. Taking into account multiple stakeholders into the design of a recommender system, while considering the central role of personalization, make the problem setting quite different from the one seen in ML. For this reason, different definitions of fairness have been introduced, allowing to have different points of view on the unfairness concerning one or many stakeholders. We can categorise two distinct groups of definitions: consumers oriented fairness (**C-Fairness**) and providers oriented fairness (**P-Fairness**) [Bur17]. The former (C-Fairness) poses the focus on disparities in recommendation quality towards the user, while the latter (P-Fairness) emphasizes the disparity in attention of items' providers.

The target, in the case of C-Fairness, is to generate recommendation scores independent from the sensitive attribute of the user [KAAS18]. Assuming $s(\mathbf{v}_u, \mathbf{v}_i)$ is the recommendation score for the interaction of user u with item i, we can express the objective of a fair strategy for the consumers as:

$$s(\mathbf{v}_u, \mathbf{v}_i) \perp a_u$$

Where a_u is the sensitive attribute of the user u and $\mathbf{v}_u$ and $\mathbf{v}_i$ the user and item embeddings.

In the case of P-Fairness, the focus in posed on the attention received by the items, grouped by different providers. In this direction, few seminal works in Information Retrieval have fostered the introduction of new metrics, introducing the notion of exposure (i.e. attention normalised by the position bias), with the objective of minimizing disparities among groups. The well-established definition of disparate exposure is the following:

Definition 7. *(Disparate Exposure). The notion of* "disparity in exposure" *quantifies the amount of attention allocated to protected subgroups in a ranking [SJ18]. Taking into account the position bias and relevance score generated through the query, the* exposure *is defined as:*

$$Exposure(G_k|\mathbf{P}(q)) = \frac{1}{|G_k|}\sum_{i \in G_k} Exposure(i_k|p_{q,i})$$

Which is the average attention received by items belonging to group G_k, dependent from the position $\mathbf{P}(q)$ of the items given the query q.

This definition is included in ML frameworks as input constraint of the algorithm, in order to avoid any correlation of the sensitive attribute with the final attention distributed among items.

3 State of the Art

3.1 Characterizing Harmful Algorithmic Curation

Multiple studies have proven how discussions on OSPs happen to be polarized and communities presenting different opinions tend to reinforce their own beliefs instead of getting connected to individuals presenting different point of views. Polarized interactions, occurring much frequently in presence of controversial topics, have an impact not only on the natural growth of the social networks, but it also affects the algorithmic output of all the ML solutions based on user activities. This effect can eventually benefit the proliferation of harmful social phenomena like echo-chambers or filter bubbles [AG05, CRF+11, BY20, LMF+07]. Different studies attempted to characterize the algorithmic effect and distinguish it from the natural interactions, and given the rapid flourishing of new OSPs and along with them new algorithmic solutions, the research area is vividly growing.

Among many, one popular approach is to audit the OSPs through fake agents or socket puppets [BAFL21, TPS+21, PZB+22]. In this direction, we mention three recent contributions, focusing on YouTube and Twitter. The first one is a computational framework, auditing the impact of algo-

rithmic curation on Twitter, showing how the tendency to show new and popular content may increase inequality in exposure of followed users [BAFL21]. In this direction, a recent work shows the filter bubbles generated by YouTube and how those can be bursted through debunking agents [TPS+21]. This study bring evidences also upon the strong presence of content sharing misinformation on the popular video sharing platform . Similar findings have been highlighted on the same platform, confirming how showing timely pseudoscientific content may redirect those on the search results page and not on the personalized sections [PZB+22].

Another relevant line of research introduces algorithmic models which, with assumptions coherent to real-world scenarios, try to explain the effect of algorithmic curation [SPGK19, CSE18, CM20a]. Particularly relevant for this manuscript a recent work [SPGK19], that proposes a modified version of the opinion dynamics model of bounded confidence, to explain how Algorithmic Bias can boost polarization among user. In a recent paper, the "filter bubble" hypothesis is included in a mathematical framework showing the impact and responsibilities OSPs have upon users, since slightly tweaking the algorithmic filtering can drammatically increase user polarization [CM20a]. A simulation study modeling the algorithmic effect [CSE18] goes in the same direction, highlighting how a recommendation algorithm can homogenize user behavior without increasing utility when generating multiple rounds of training. The authors also define the effect of algorithmic confounding, expressed as the platform attempts to model user behavior without accounting for recommendations.

3.2 Mitigating Algorithmic Polarization

A significant effort has been devoted also for mitigating negative effects produced by the interaction between polarized communities and algorithmic recommendations. The proposed algorithms are based on the tradeoff between the relevance of the outcome and the metric, to optimize to reduce the harmful effect of the algorithm. Approaches modifying the al-

gorithmic effect either suggest non-harmful contents to the users or proposes to augment the underlying network topology.

We introduce here few recent promising methods for the first class of algorithms, while for a discussion on methods modifying directly the network topology, we direct the user to Section 6.2.

Recently a bandit algorithm has been proposed to reduce polarization in personalized recommendations, by allowing the user to constrain the distribution from which content is selected [CKSV19]. Also, another approach proposes to reduce polarization through antidote data included in the input, to improve the social desirability of recommender system outputs [RGC19]. Finally, the work by [TRG21] introduces FRediECH, an echo chamber-aware friend recommendation approach that learns users and echo chamber representations from the shared content and past users' and communities' interactions.

3.3 Studying Algorithmic Discrimination in OSPs

Evidences of discrimination through ranking and recommendation methods raised society awareness on the responsibility OSPs have on the users involved in an ML lifecycle. Ranking systems are technologies with a direct impact on the searcher (final user) and the searched (final output) [1]. The algorithmic harmful effects are studied either through a black-box approach, i.e. auditing the online platforms without having access to the underlying algorithms, or through a white-box approach, which the target is to reproduce from scratch the solution implemented within OSPs.

A seminal contribution in this area comes from [MAD+17], showing evidences the role of user demographics over differential satisfaction in search engines queries. The study finds significant differences in usage patterns and evaluation metrics for different users' groups based on age and gender.

After that contribution, two macro-areas of research have flourished around two applications: automated hiring systems (AHSs) and the role

[1] "gender bias" evidences from https://www.bbc.com/news/newsbeat-32332603

of their vendors [WGJ+21, SMDE20, RBKL20]; online housing markets and the role of different stakeholders [AES+20, SFC+21].

In the case of AHSs, we can identify Pymetrics as one of the main vendors, established as one of the firsts to include ML fair algorithms into the pipeline to recommend job candidates to companies. The first work by [WGJ+21] audits the platform to then proposes recommendations for the vendors, including transparency, introduction of new legal standards shaping statistical tests and more inclusive de-biasing techniques. The practices of Pymetrics and similar platforms have been challenged also by [SMDE20], which emphasize how AHSs frameworks, in the current state, are unfeasible and far from the socio-legal context of the UK, and more generally not oriented towards EU regulations. Another work, in a similar spirit, investigates both technical and legal perspectives of AHSs. The focus in this contribution is posed on the risks and trade-offs faced by the vendors, emphasizing also in this case how algorithmic de-biasing techniques need to cope with anti-discrimination laws [RBKL20].

Online housing markets are closely related to intimate platforms, in which listers (who is renting/selling) and seekers (who is renting/buying) try to find a mutual match, which can be highly driven by demographics or other sensitive information [HTBL18].

Among the first ones, the work by [AES+20] analyzes seekers' perception upon online advertising and search-result ranking in different housing portals in the U.S. Their results show evidences of discrimination against gender and geographic location reflected in the ranking of properties. In the same direction, our recent work shows how the recommendation algorithm differs in performances across groups, further reducing opportunities of finding opportunities for some minorities in both sides, either as a seeker or a lister. Among the aforementioned, this study is the only one not belonging to the black-box cases, since we have access to the underlying algorithms running within the online platform.

In the case of white-box approaches, many are the domains in which the bias propagating through a recommender system has been analysed, such as, MOOCs recommendation [BFM19], books suggestions [ETA+18] and next song prediction [AFBS21]. This line of work involves the em-

pirical evaluation of biases involving both consumers and providers, in relationship to other phenomena like popularity bias and catalogue size [ETA+18, AFBS21, ESM19]. One influential work by [ETA+18] analyses the effect of age and gender on the utility of the recommendations produced through publicly available datasets in the case of collaborative filtering methods. The paper shows also how there is no evidence about the influence of the relative sizes of users' group on the recommendations. In our recent work we find evidences in contrast with the previous findings, at least for the relative sizes of groups of providers [AFBS21]. We show how, in the case of session-based RS, a music artist presenting a popular but small catalog may receive a reduced amount of exposure if compared to artists presenting a popular but big catalog.

3.4 Mitigating Discrimination in RS

In this section we present the most relevant methods mitigating algorithmic discrimination while adopting ranking-based ML solutions, with a particular emphasis on RS. Methods proposed in the literature of IR to fight algorithmic discrimination pose the focus on the attention received by the searched elements [ZYS21]. We can distinguish between approaches reducing the observed unfairness (e.g. disparate exposure) and the ones questioning the probabilistic law which generated the rankings.

In the first line of research, we can find similar contributions aiming at re-allocating attention among searched items including a fairness definition coherent with the context (e.g. dataset and application) [ZC20, SJ18]. While, when proposing probabilistic solutions, those can be distinguished based on the assumptions on data characteristics, such as the probability law or the input level of bias [CMV20, GSB21]. For an extensive literature review on Algorithmic Fairness and IR methods, we point the reader towards a survey by [ZYS21] and a critical review of fair rankings [PPM+22].

Harmful biases in RS can be mitigated at three different stages of the

recommendation pipeline: (i) sanitizing the input data (*pre-processing*); (ii) modifying the model generating the final output (*in-processing*); modifying the final output generated through the original recommender (*post-processing*).

One of the first lines of work connects with the criteria of statistical independence, described in the previous chapter, popular in settings of supervised classification. The work by [KAAS18] is the first one introducing the concept of recommendation independence, proposing a in-processing algorithm which generates embeddings independent of the sensitive attribute associated to the user. Another seminal work [ZHC18], with the aim of reducing unfairness for the consumers, introduced a tensor-based recommendation algorithm. This new formulation, with the assumption that the latent features are tainted by the bias of demographics, allows to isolate and extract those sensitive features through adding regularizer in the loss function, sanitizing the *"sensitive dimensions"*.

A similar approach has been proposed also for settings unexplored before, such as graph embeddings and reputation-based RS [BH19, RB20]. In the former, the network embeddings are generated independent of the intersection of multiple sensitive attributes associated to the node [BH19]. The independence is reached through an adversarial loss which the objective is to build embeddings orthogonal to sensitive features of the node. While in the latter the recommendation independence is treated as a constraint in the input data, introducing the measure of *disparate reputation* [RB20].

Another popular line of research connects with the criteria of calibration of predictions in ML. In particular, the objective of the mitigation strategies in this area try to redistribute the outcome of the output, in a fair manner with respect to some data characteristics. A seminal work is the one by [Ste18], which the objective is to optimize the KL-divergence between distribution of movies genres in recommendation and input data for each user. In a similar spirit, a paper by [YH17] attempts to equalize bias in preferences among the group of users before and after the recommendations. They propose new fairness metrics optimized in-processing, based on two forms of underrepresentation: population imbalance and

observation bias.

Calibration of recommendation equalizing fairness among stakeholders is also performed through post-processing techniques. For example, a seminal paper by [LB18] considers P-fairness in the Kiva.org platform, which grants loans to low-income entrepreneurs. They propose a re-ranking function (based on xQuad), which balances recommendation accuracy and fairness, by dynamically adding a personalized bonus to the items of the uncovered providers [CNPS11]. Finally, a recent work [MMB$^+$18] connects the fairness of the providers with the level of popularity. More specifically, the work shows through re-ranking, how it is possible to define, in a two-sided marketplace, personalized fairness definitions considering the user tolerance towards more fair contents. Artists are divided in ten bins based on their popularity, and a fairness metric that rewards recommendation lists that are diverse in terms of popularity bins is defined. Several policies are defined to study the trade-offs between user-relevance and fairness, adapting the level of fairness depending on user tolerance.

Part II

Bias in People Recommender Systems

4 The Effect of Homophily on Exposure in People Recommender Systems

4.1 Introduction

People recommender systems, also known as contact recommenders or *who-to-follow* link recommenders [GP16, SCC18], suggest to users possibly relevant new connections. These algorithms are a core functionality of every social media platform, as they contribute to stimulate new interactions, ultimately affecting the growth of the network [SSG16]. As such, they can play a key role in building the "social capital" of individuals (e.g., their number of followers). Besides general-purpose social networking platforms, people recommenders are also widely used to suggest connections between users in other environments, such as employment services [HKW+14, LOR+16b, LOR+16a, HVR+16, DMP+16], educational services [VMG16, ZML+16], co-workers suggesting [GRW09] or expert finding [HLT16, SD13, GAC+13].

It is thus of great importance to study potential algorithmic bias that might lead to disparate exposure of individuals. For instance, [SSG16] analyzed the abrupt changes in Twitter's network structure after the introduction of the "Who to Follow" feature, and found that users across the popularity spectrum benefitted from the recommendations; however, the

most popular users profited substantially more than average. Similar findings were reported by [DGM10], who conducted a large-scale user study on IBM's Social-Blue social network site. While these two works focus on the inequalities at the level of individual users, some authors have analysed a *glass ceiling*[1] effect for women in social networks [NGL+16]. For instance, a recent work by [SRC18] investigates the role of gender in organic and artificial growth of social networks, using a large social graph from Instagram, where women are the majority class. Their theoretical model predicts a glass ceiling at the expense of a minority, but their empirical observations show glass ceiling against the female majority. They explain this apparent contradiction by the different level of homophily of the two groups.

In this chapter, *we provide a systematic analysis of the effect of homophily on disparate exposure of minorities in people recommender systems*.

Homophily, the tendency of people to connect with others who are similar to them, is one of the main driving forces behind the organic growth of a social network, thus strongly influencing the main input of people recommender systems, i.e., the structure of the network. The next toy example shows the potential effect of homophily on the recommendations provided by an algorithm.

Example. Figure 4.1 reports two cases. In both cases, the starting social graph is composed of ten nodes: 7 in the majority group (blue), and 3 in the minority group (red). However, in the bottom case the minority exhibits a stronger level of homophily: users belonging to the minority (red) group tend to connect among themselves more than the ones in the network on the top case (a more formal definition of homophily will be given in Section 4.3). We assume a "preferential attachment" recommender, which suggests to a generic node u as node to follow, the one with the highest number of followers from the set of nodes at distance 2, i.e., nodes followed by her neighbors. The networks in the center column contain the recommendations produced, where the color of an edge is the

[1]This is a metaphor referring to a sort of invisible barrier that prevents a group of people from rising in a hierarchy.

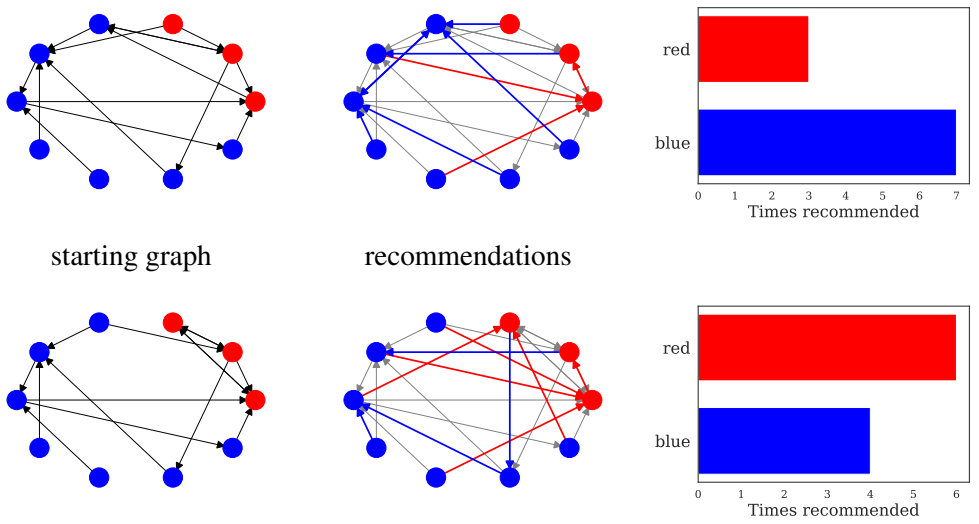


Figure 4.1: Example depicting the role of homophily in a recommender system. The social graphs on the left are composed of ten nodes: 7 in the majority group (blue), and 3 in the minority group (red). The graphs are directed: a link (u, v) indicates the fact that u *follows* v. The graphs in the center reports the links recommended using the color of the node which is recommended to be followed.

same as that of the node who gets recommended to the source user. It is evident that homophily allows minorities to get much more exposure with respect to a less homopilic scenario (i.e., in the bottom network of the mid column, the number of red edges has increased with respect to the one above it, while that of the blue nodes has decreased).

Chapter contributions and roadmap. In this chapter, we characterize the exposure given by different recommendation algorithms to different groups of users, as a function of their relative *sizes* and the *homophily* of each group. We perform experiments with both real-world social networks, with groups defined by sensitive features such as gender or age, and synthetic graphs where we can explore different combinations of majority/minority sizes and homophily. This study sheds light into phenomena that suggest we must measure and mitigate negative effects of recommender systems, including user discrimination and unfairness [EBD19] and a network's possible lack of resilience [GMS13]. Specifically, our work makes the following contributions:

- We provide a systematic study of the disparate exposure produced by contact recommendation algorithms, on real social networks and on synthetic datasets;

- We show that homophily plays a key role in the exposure given to different groups; when the minority is homophilic, there is a disparate exposure in favor of the minority class; when the minority is not homophilic, the disparate exposure is in favor of the majority class;

- Consistently with the literature, our analysis shows that recommenders amplify the rich-get-richer phenomena, thus introducing inequality of exposure. Such observed inequality, however, is stronger within the minority class compared to the majority class, especially when the minority is homophilic. This is explained by the fact that the minority class is over-represented in the sub-population of most recommended nodes when the minority is homophilic, and under-represented when the minority is not homophilic;

- We show that, when taking into account the initial in-degree, nodes in the minority class are disadvantaged in terms of exposure, regardless of the homophily of the minority class. In other words, among nodes with similar in- degree, the ones that belong to the majority class are likely to be recommended more;

- Finally, we show that the relative size of the minority does not impact exposure as much as homophily does.

The rest of the chapter is structured as follows. Section 4.2 discusses related work. Section 4.3 introduces the metrics and algorithms we considered. Section 4.4 presents the experiments performed on real graphs and Section 4.5 those on synthetic graphs. Finally, Section 4.6 presents our conclusions.

4.2 Related Work

In of the first work, a large-scale proprietary dataset is analyzed, containing a complete snapshot of Twitter and its "Who-To-Follow" recommender [SSG16]. Specifically, they study user behavior before and after the introduction of the recommender system in this social platform. They found a faster in-link growth for popular nodes, with a sub-linear popularity effect. In contrast with our work, user demographics were not taken into account and consequently, the role of homophily was not considered. We also consider more than one algorithm and measure the effect of the recommender at both the individual and the group level. Our study suggests that node popularity (in our case, represented by the in-degree) is not the only crucial factor needed to characterize the rich-the-richer phenomenon, since popular nodes are treated differently according to the group they belong to (i.e., majority or minority) and the level of homophily in a group.

A recent work by [DGM10] investigated how recommendations can affect the global and local structure of a network. They focused on differences in topological features such as degree distribution skewness and

node betweenness. In contrast, in our study we consider more properties of the nodes (such as the group they belong to and the exposure they obtain), in addition to characteristics such as node degree that have been previously studied. Another contribution was able to prove a *glass ceiling* effect in social networks [NGL+16]. They investigated how perceived gender and online exposure can be linked, showing that users perceived as female experience a "glass ceiling" effect, similar to the one that makes it harder for women to reach higher positions in companies. This study was a seminal work around discrimination in social media interactions, which exaggerates stereotypes present in society. Our work tries to go in-depth into this phenomenon, trying to understand how network interactions along with recommendation algorithms might lead to disparate exposure of minority groups (e.g., how homophily affects the generated recommendations).

Through a study using synthetic data, [LKJ+17] analyzed the characteristics of minorities of different sizes in a bi-populated graph, introducing homophily in a network growth process. We extend the model they proposed to perform analysis of recommendation algorithms on synthetic data.

Recently, a work by [KGW+18] studied disparate effects introduced by homophily, such as disparities in ranking distribution over subgroups, but without investigating its consequences on recommendations. This work strongly motivates ours, showing the research gap related to recommender systems effects.

A recent work by [SRC18] investigate the role of gender in organic and artificial growth of social networks, using a large social graph from Instagram, where female are the majority class. Their theoretical model predicts a glass ceiling at the expense of a minority, but their empirical observations show glass ceiling against the female majority. They reconciliate this apparent contradiction by extending their theoretical model in order to keep in consideration the different level of homophily of the two groups: in particular, a homophilic minority can flip the glass ceiling effect to come at the expense of the majority. Our systematic analysis confirms this intuition.

Related to our findings is also the *few-get-richer effect* phenomenon, which explains how the minority class tends to be top-ranked by popularity-based systems. This phenomenon has been analytically proven by a recent work by [GGM19] and, although not embedded in the algorithms we considered, it finds empirical evidence in our experiments.

4.3 Preliminaries

We consider a bi-populated and directed social network, represented as a graph $G = (V, E, c)$ where V is the set of nodes, $E \subseteq V \times V$ is the set of directed edges, such that an edge $(u, v) \in E$ indicates the fact that u *follows* v, and $c : V \to \{V_1, V_2\}$ is a function assigning each node to one of two classes V_1, V_2 which partition V. We denote by s_1 the fraction of nodes belonging to the first class (i.e., $s_1 = |V_1|/|V|$) and by s_2 the fraction of nodes belonging to the second class (i.e., $s_2 = 1 - s_1$).

We also consider a people recommender system represented by a function $\ell : (V \times V) \setminus E \to [0, 1]$, which associates to each non-existing edge (u, v) a score $\ell(u, v) \in [0, 1]$. From a probabilistic standpoint, $\ell(u, v)$ can be interpreted as the probability for such recommendation to create a new connection that is accepted by u. In each round of recommendation, the system recommends to each node $u \in V$ a set $R(u)$ of other nodes to follow, where $|R(u)| = k$, for a given parameter $k \in \mathbb{N}^+$. Typically, $R(u)$ will contain top-k nodes v w.r.t. $\ell(u, v)$.

Exposure. In this work, we consider one single round of recommendation and focus on how many times each node v appears in the recommendation sets of all the other nodes. We call this quantity *exposure* of v and denote it

$$\psi(v) = |\{u \in V | v \in R(u)\}|.$$

In particular, we are interested in the fraction of recommendations that each of the two classes of nodes, V_1 and V_2, receives. The exposure of a

specific subgroup i can be expressed as:

$$\mathcal{V}_i = \frac{1}{k|V|} \sum_{v \in V_i} \psi(v) \tag{4.1}$$

Disparate Exposure. Considering the size of the two groups inside the graph, we can also refer to them as minority m and majority M, which respectively present relative size s_m and s_M. Then, the simplest way for defining differences in exposure between those two groups of users, used in ranking systems [SJ18], is overall exposure normalized by group size, namely:

$$\Delta(\mathcal{V}) = \frac{\mathcal{V}_m}{s_m} - \frac{\mathcal{V}_M}{s_M} \tag{4.2}$$

We call this measure *disparate exposure*: this measure ranges in $[-\frac{1}{s_M}, \frac{1}{s_m}]$ and it is zero when the exposure (fraction of recommendations) received by the minority is equal to the relative size of the minority. Therefore, a disparate exposure close to zero represents a situation in which no group is favored, large negative values indicate the minority class is given a disproportionately large exposure, and large positive values indicate the majority class is given a disproportionately large exposure.

Homophily. Homophily is a well-known phenomenon in network science and can be expressed as *the tendency of people to connect to similar people*, or in our case, of people in a group to connect to people in the same group. We measure homophily with respect to a random configuration, inspired by work analyzing dyadicity in signed networks [PB07]:

$$h_i = \frac{|E_{ii}|}{|E_{i.}|} - s_i \tag{4.3}$$

where $E_{ii} = \{(u, v) \in E | u \in V_i \wedge v \in V_i\}$ and $E_{i.} = \{(u, v) \in E | u \in V_i\}$. This measure expresses the difference between the number of observed intra-group edges and the expected number if edges were created

at random. It ranges in the interval $(-s_i, 1 - s_i]$. A group is called homophilic if the tendency to connect to nodes of the same group is stronger than expected ($h_i > 0$), heterophilic when this tendency is weaker than expected ($h_i < 0$), and neutral if the number of edges towards each group is consistent with the proportion of nodes in each group ($h_i = 0$).

Recommendation algorithms. We consider four different methods for link recommendation and investigate the node exposure generated by those. One is a baseline random recommender, and the other three are state-of-the-art algorithms, representative of three distinct families of methods (based on topology, random walks, and collaborative filtering), that we have chosen because of their popularity and performance [LFS17b, SC18].

ADA: Network Topology Based. Among the different heuristics which aim to define similarity between nodes looking at the graph topology, we select the *Adamic-Adar coefficient* (for short "ADA" in the rest of the chapter), method that penalizes connections with high degree nodes.

SLS: Random Walks Based. As representative of random-walks based approaches, we use SALSA (Stochastic Approach for Link-Structure Analysis) ("SLS" in the rest of the chapter), which is at the basis of the *who-to-follow* recommender at Twitter [SSG16]. Recommendation of a generic link is defined as the probability of the source node to jump to the target one, rather than to any other node in the graph.

ALS: Collaborative Filtering Based. Connections among nodes can be considered as implicit feedback in a collaborative filtering approach. An Alternating Least Squares algorithm ("ALS" in the rest of the chapter) is selected to perform recommendations [HKV08]. New links are suggested based on latent features extracted from the adjacency matrix.

RND: Random baseline. As baseline, we consider a random recommender ("RND" in the rest of the chapter), which picks recommendations uniformly at random from the candidate nodes at distance 2.

Aligned with the common practice among social network providers, such as Facebook[2] and Twitter[3], which suggest users with mutual connections, recommendations in our experiments are chosen from the set of missing links at distance two (friends of friends).

4.4 Observations on Real-World Graphs

In this section, we analyze data from two social networking sites, exploring how the role of homophily on groups of nodes can play a role in the generation of recommendations. We remark that this experimentation is made difficult because there are very few social networking datasets where nodes can be partitioned into classes based on demographic attributes.

4.4.1 Datasets

TUENTI. Known as the "Spanish Facebook," Tuenti has been a popular social networking site in Spain. The data we use includes some demographic information about users [LVKK16].

Nodes are users and edges are defined by wall-post interactions, i.e., a user posting on another user's "wall." Specifically, a directed edge (u, v) exists if user u posted at least t times on the wall of a user v. To remove sporadic interactions, we consider $t = 3$ as a threshold. This network has 8,983,560 nodes (users) and 17,830,103 edges.

In order to have a fair comparison of the performance with different datasets, we decided to create samples of equal size. Finally, the sample

[2]https://www.facebook.com/help/163810437015615

[3]https://help.twitter.com/en/using-twitter/account-suggestions

size was set to 500,000 nodes, for computational reasons and due to the large amount of experiments we performed. To sample, we follow the work by [WSK+17] and use a random walk based algorithm, which has been shown to preserve characteristics that are of interest in our analysis, such as the relative sizes of minority and majority classes, as well as their level of homophily. The resulting network contains 500,000 nodes and 2,813,744 edges.

Next, we create different bi-populated networks using different partitions by gender and age, whose basic characteristics are reported in Table 4.1 and Figure 4.2 (in the table and figure, datasets are ordered by decreasing homophily of the minority):

- **TUENTI-G** is the network partitioned by gender. It is characterized by an absence of homophily in both groups and, among the three partitions of the original dataset, it is the one with the largest minority class (females, $s_m = 0.39$).

- **TUENTI-A16** is the network partitioned by age with 16 as cut-off. This dataset presents two groups which are both homophilic, with a smaller minority than the previous case (younger than 16, $s_m = 0.30$).

- **TUENTI-A30** is also based on a partition by age with 30 as cut-off. It presents a very small minority (older than 30, $s_m = 0.04$) and it lacks homophily in both groups.

POKEC. This is a popular social networking site in Slovakia. The data is publicly available[4], anonymized, and includes some demographic information.

In total, the network contains 1,632,640 nodes (users) and 22,301,602 edges, where each edge represents a "follow" relationship, which can be

[4]https://snap.stanford.edu/data/soc-Pokec.html

Name	Attribute	$\|V\|$	$\|E\|$	s_m	h_m	h_M
TUENTI-A16	age	500000	2813744	0.30	0.42	0.14
POKEC-A21	age	500000	8635662	0.46	0.34	0.19
TUENTI-A30	age	500000	2813744	0.04	0.08	0.02
TUENTI-G	gender	500000	2813744	0.39	0.02	0.07

Table 4.1: Characteristics of real-world social networks analyzed: dataset name, attribute used for partitioning, number of nodes, number of edges, proportion of the minority size, homophily of the minority, and homophily of the majority.

non-symmetrical. We adopt the same sampling approach used for Tuenti and produce a network containing 500,000 nodes and 8,635,662 edges.

We create the two classes of nodes by partitioning by age with a cut-off of 21. The resulting dataset, dubbed **POKEC-A21**, presents quite well-balanced groups (minority is younger than 21, $s_m = 0.46$), with the minority more homophilic than the majority.

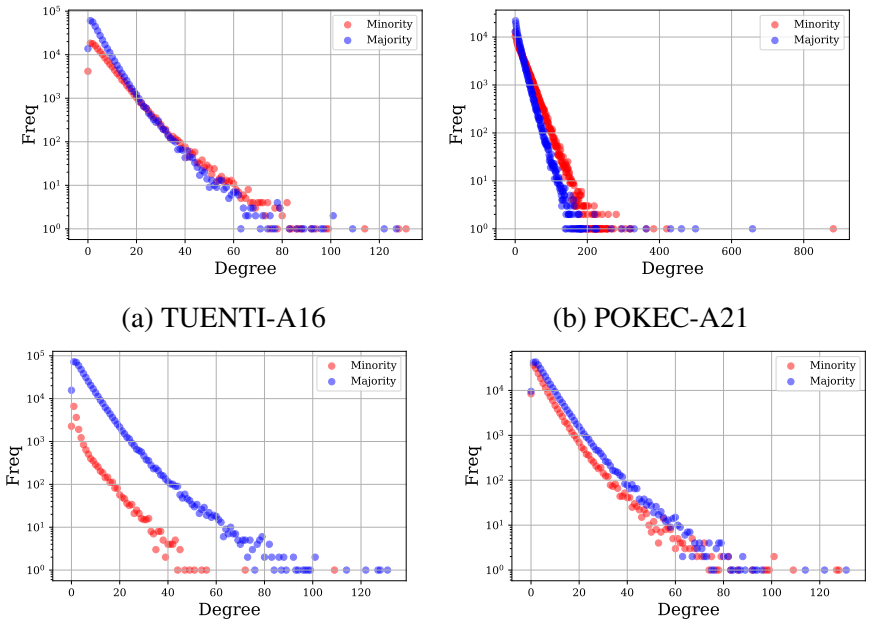
Figure 4.2 reports the in-degree (number of followers) distribution of the minority and majority classes in each social network. We can observe that in the datasets with a homophilic minoritiy (TUENTI-A16 and POKEC-A21), the minority class exhibits an advantage in terms of high in-degree nodes.

4.4.2 Disparate Exposure

We next apply the four link recommendation methods to all our networks, recommending to each node $k = 5$ other nodes; then we measure exposure, i.e., how many times each node appears in the recommendations to other nodes. Table 4.2 reports disparate exposure $\Delta(\mathcal{V})$ between the minority and majority class, defined as in Eq. 4.2: a value of $\Delta(\mathcal{V}) > 0$ indicates that the minority class is favored in terms of exposure, while $\Delta(\mathcal{V}) < 0$ indicates that the majority class is favored. A first observation we can draw is the following:

Network	Method	$\Delta(\mathcal{V})$	$\Delta(\mathcal{V}_{<q_{90}})$	$\Delta(\mathcal{V}_{<q_{80}})$	$\Delta(\mathcal{V}_{>q_{90}})$	$\Delta(\mathcal{V}_{>q_{80}})$
TUENTI-A16 $s_m = 0.3$ $h_m = 0.42$	ALS	0.517	0.184	0.086	0.681	0.630
	SLS	0.264	0.069	0.014	0.464	0.396
	ADA	0.134	0.071	0.048	0.249	0.209
	RND	0.149	0.155	0.154	0.119	0.123
POKEC-A21 $s_m = 0.46$ $h_m = 0.34$	ALS	0.900	0.645	0.401	0.985	0.944
	SLS	0.571	0.312	0.196	0.731	0.653
	ADA	0.328	0.259	0.208	0.434	0.386
	RND	0.310	0.322	0.309	0.285	0.282
TUENTI-A30 $s_m = 0.04$ $h_m = 0.08$	ALS	-0.276	-0.397	-0.433	-0.224	-0.306
	SLS	-0.350	-0.446	-0.504	-0.251	-0.328
	ADA	-0.359	-0.436	-0.501	-0.200	-0.273
	RND	-0.333	-0.423	-0.503	-0.105	-0.197
TUENTI-G $s_m = 0.39$ $h_m = 0.02$	ALS	-0.264	-0.323	-0.292	-0.267	-0.178
	SLS	-0.291	-0.348	-0.324	-0.261	-0.200
	ADA	-0.212	-0.252	-0.235	-0.157	-0.122
	RND	-0.149	-0.186	-0.168	-0.086	-0.062

Table 4.2: Disparate exposure ($\Delta(\mathcal{V})$ introduced by different recommenders: $\Delta(\mathcal{V}_{<q_{90}})$ and $\Delta(\mathcal{V}_{<q_{80}})$ refers to the same measure when removing the top-10% and top-20% of in-degree nodes, respectively, from each class; while $\Delta(\mathcal{V}_{>q_{80}})$ and $\Delta(\mathcal{V}_{>q_{90}})$ refers to the measure computed on the top-20% and top-10% in-degree nodes of each class.



(a) TUENTI-A16

(b) POKEC-A21

(c) TUENTI-A30

(d) TUENTI-G

Figure 4.2: In-degree (number of followers) distribution of the minority and majority classes in each social network. We can observe that in the datasets with a homophilic minoritiy (TUENTI-A16 and POKEC-A21), the minority class exhibits an advantage in terms of high in-degree nodes.

Observation 1. *In graphs with a homophilic minority, there is a disparate exposure in favor of the minority class. When the minority is not homophilic, the disparate exposure is in favor of the majority class. This holds for all the link recommendation methods we tested.*

Although the observation above holds true regardless of the recommender we tested, we observe that the effect is more evident with the two more sophisticated methods, ALS and SLS. For instance, in the POKEC-A21 dataset, with a minority almost as large as the majority ($s_m = 0.46$),

a homophilic minority ($h_m = 0.34$) and a slightly homophilic majority ($h_M = 0.19$), the ALS recommender gives high exposure to the minority ($\Delta(\mathcal{V}) = 0.9$).

We conjecture that this result might depend on the fact that, when in presence of a homophilic minority, the minority class presents more nodes with high in-degree than the majority. Thus Table 4.2 also reports what happens when we exclude the top-20% (column $\Delta(\mathcal{V}_{<q_{80}})$) and the top-10% (column $\Delta(\mathcal{V}_{<q_{90}})$) high in-degree nodes from each of the two classes.

As expected, when we remove hubs from the analysis, the disparate exposure in favor of the minority class in the datasets with homophilic minority (TUENTI-A16 and POKEC-A21) gets reduced substantially. This is confirmed by the columns $\Delta(\mathcal{V}_{>q_{80}})$ and $\Delta(\mathcal{V}_{>q_{90}})$ which focus only on the top-20% and the top-10% high-degree nodes, for which the disparate exposure in favor of the minority is very high. Of course this does not hold for the RND recommender, which depends much less on the in-degree of the nodes than the other recommenders.

When considering the TUENTI-G network partitioned by gender, the size of the minority is much smaller than that of the majority ($s_m = 0.39$), and both groups are characterized by neutral homophily (neither homophily nor heterophily). Under this setting, the distribution of exposure harms the minority. For nodes with highest degree, the effect is mitigated, but still indicating that minority nodes are receiving slightly less exposure than what should correspond to them given their degree. Consequently, excluding nodes with highest degree, the difference in exposure is even stronger, showing that minority long-tail nodes are at a disadvantage when compared to their peers in the other group.

TUENTI-A30 is characterized by the smallest minority ($s_m = 0.04$), and absence of homophily in both majority and minority groups. Under this setting, similarly to what happened in the TUENTI-G network, the minority receives less exposure (even more than in the TUENTI-G network). Also in this case, the effect is slightly mitigated when looking at the nodes in the top of the in-degree distributions and exaggerated for the rest of the graph.

Observation 2. *The hubs existing in the minority group receive large exposure. In contexts in which the minority is homophilic, this exaggerates the disparate exposure in favor of the minority. In contexts in which the minority is not homophilic, this helps slightly mitigate the disparate exposure against the minority.*

This last consideration motivates further investigation of the interplay between in-degree, exposure, and the belonging to the minority or the majority class.

4.4.3 Rich-get-richer Effect

We next study inequality of exposure of nodes within each of the two classes. Figure 4.3 reports Lorenz Curves[5] of exposure of nodes (ψ) and in-degree (denoted as d_{in}) inside the two subgroups. Lorenz Curves are a popular graphical tool to show the cumulative distribution of a variable inside a population, emphasizing the differences with respect to a hypothetical random distribution. They are widely used to evidence inequality in wealth distribution among countries or more generally, comparing the wealth distribution of subpopulations [Cha12].

These plots present on the x-axis the percentile of the population and on the y-axis the fraction of cumulative distribution of the wealth. For instance a point (0.8, 0.2) indicates that 80% of the population has 20% of the wealth. In case of absolute inequality, all the wealth is assigned to only one person and the line correspond to the x-axis. In the case of perfect equality, the wealth is distributed equally along the sample, corresponding to the $x = y$ diagonal. In Figure 4.3, the "wealth" corresponds to the in-degree (denoted as d_{in}) of nodes and to their exposure (ψ) with respect to the recommendations produced by the ALS and SLS methods inside the two classes. We report only two methods for sake of space, the other methods produce similar results.

[5]https://en.wikipedia.org/wiki/Lorenz_curve

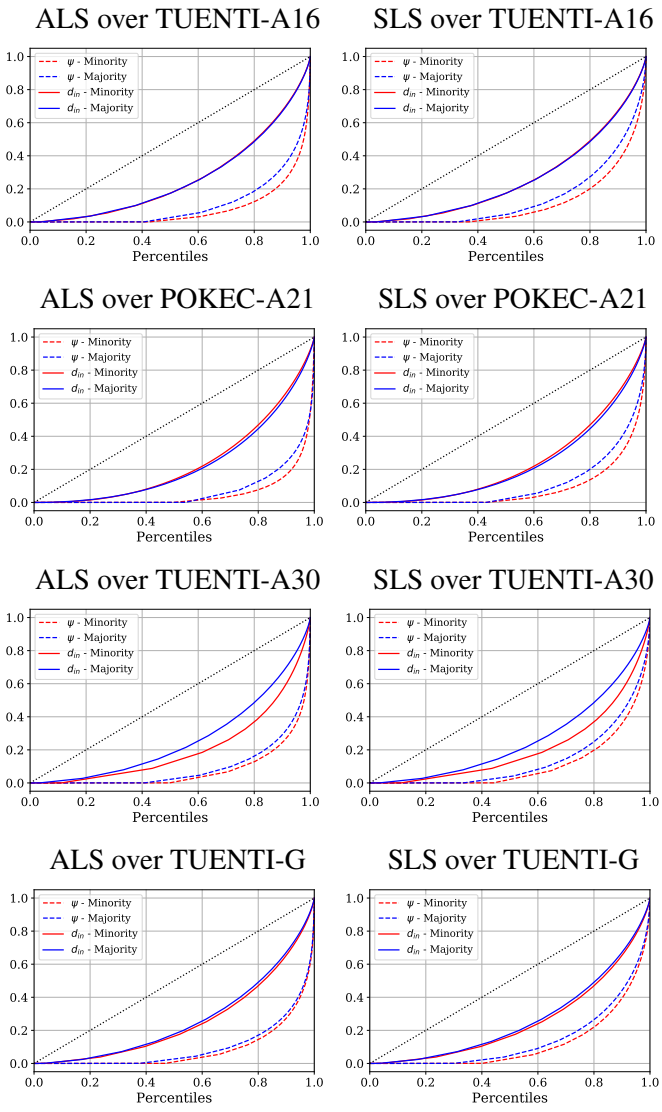


Figure 4.3: (Best seen in color.) Lorenz Curves depicting inequality. Dashed lines represent recommendations, solid lines represent in-degree. The minority is in red, the majority in blue. Recommendations introduce more inequality than the degree distribution, and this inequality is stronger in the minority class.

The first (well-known) observation on Figure 4.3 is that link recommenders amplify the intrinsic "inequality" of the in-degree distribution, as shown by the difference existing between the solid lines and the dashed lines. This rich-get-richer effect is innate in the link recommendation task, thus not surprising.

Instead, more surprising is the fact that such effect is stronger within the minority than within the majority class (difference between the dashed blue line and the dashed red line) and this is consistent among all datasets and all recommenders, although being more evident in datasets with a homophilic minority. This confirms what we observed in Table 4.2, i.e., the fact that there are a few hubs that receive most of the exposure in the minority class.

Observation 3. *Recommenders amplify the rich-get-richer phenomena observed for in-degree, thus introducing more inequality of exposure. Such observed inequality is stronger within the minority class compared to the majority class, especially when the minority is homophilic.*

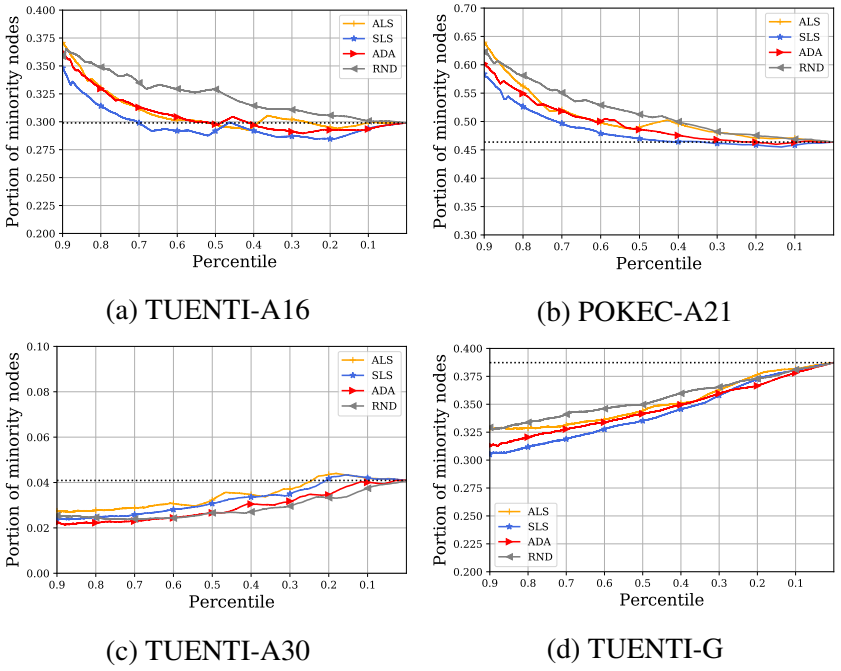
4.4.4 Most Visible Nodes

We investigate further these observations by showing the fraction of nodes of the minority class that belong to the most visible nodes. Figure 4.4 reports the fraction of nodes belonging to the minority class that are among the most visible ones on each dataset and for each recommender. For instance, in the left-most point of Figure 4.4(b) we can see that in the POKEC-A21 dataset, while the minority class represents 46% of the population, it rises to 58-65% (depending on the recommender) when checking only the 10% of most visible nodes. A similar observation holds for the other graph with homophilic minority (TUENTI-A16).

However, on graphs in which the minority is not homophilic, the trend is completely overturned. For instance, in the TUENTI-G dataset (Figure 4.4(d)) while the minority class represents 39% of the overall population, when focusing only on the sub-population of the 10% most visible

nodes, the minority class is under-represented: i.e., 32-37% (depending on the recommender).

Observation 4. *The minority class is over-represented in the sub-population of most recommended nodes when the minority is homophilic, and under-represented when the minority is not homophilic.*



(a) TUENTI-A16

(b) POKEC-A21

(c) TUENTI-A30

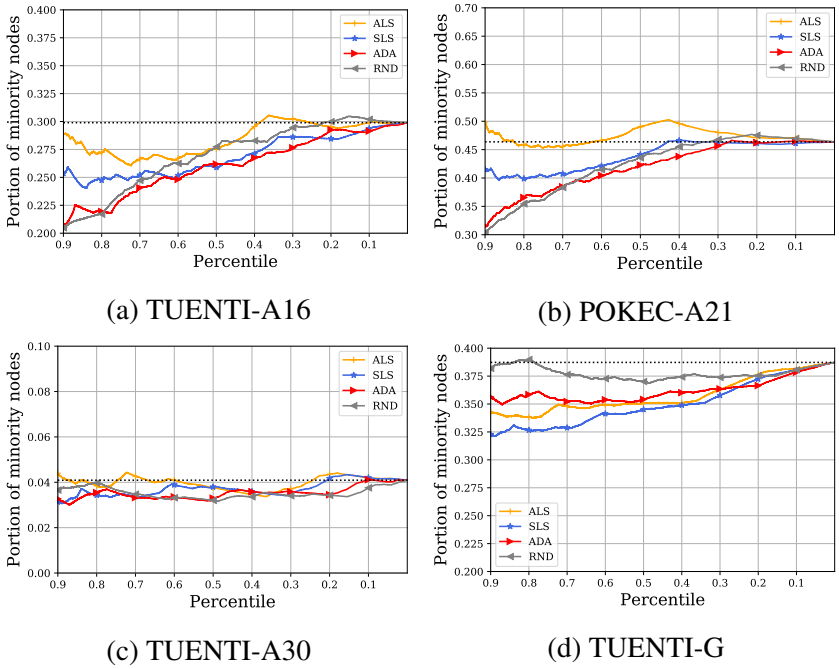
(d) TUENTI-G

Figure 4.4: Portion of the minority class in the top nodes, sorted by ψ.

Most of these observations seen so far are rooted in the fact that in the datasets with a homophilic minority (TUENTI-A16 and POKEC-A21), in-degree influences differences in exposure distribution. However, it is interesting to ask whether two nodes with similar in-degree, one from the minority and one from the majority class, have similar exposure.

4.4.5 Individual Fairness

We now adopt an individual fairness standpoint, i.e., the principle according to which similar individuals should receive a similar treatment [DHP+12]. In our setting, being similar means having similar in-degree (e.g., a similar number of "followers" in a social networking site). Therefore, we sort nodes by ψ/d_{in}, i.e., the number of times a node is recommended divided by its in-degree.



(a) TUENTI-A16

(b) POKEC-A21

(c) TUENTI-A30

(d) TUENTI-G

Figure 4.5: Portion of the minority class in the top nodes, sorted by ψ/d_{in}.

Figure 4.5 shows that, contrarily to what is seen in Figure 4.4, if we normalize by in-degree then nodes in the minority group are underrepresented among top nodes, regardless of the level of homophily of the minority. For instance, in all graphs and all recommenders, if we take the top-40% nodes by ψ/d_{in}, then the fraction of nodes belonging to the minority class is always below the dotted line, which represents the relative

size of the minority in the network.

Observation 5. *When taking into account in-degree, nodes in the minority class are disadvantaged in terms of exposure, regardless of the homophily of the minority class. In other words, among nodes with similar in-degree, the ones that belong to the majority class are likely to be recommended more.*

4.5 Observations on Synthetic Graphs

Synthetic networks allow us to test the extent to which the observations on real-world graphs hold for a wider range of configurations: in particular, they allow us to control the level of homophily in the two groups and the relative size of the minority (which would be impossible to do on real-world graphs).

We next discuss how we generate syntectic networks.

4.5.1 Data Generation Process

Our goal is to generate bi-populated directed networks where we can control the homophily of each of the two groups and their relative size. This is a non-trivial task. Our solution is inspired by the *Biased Preferential-Attachment* model introduced for undirected graphs [LKJ+17], and that we extend to produce directed graphs.

Under our model, the tendency to connect to other nodes is regulated by in-degree distribution and *in-process homophily*. The latter, which represents for each group the tendency to connect to same peers along the process, is indicated by ρ, which is a non-negative coefficient bounded in the interval $(0, 1]$. Nodes are partitioned into a minority m and a majority M, where a generic node v is associated to the minority m with probability p_m and to the majority M with probability $p_M = (1 - p_m)$. In the long run, these two probabilities correspond to the fraction of nodes belonging to the two partitions. The value of ρ depends on the class of

the source node, i.e., assuming u as new node to add with $c(u) = m$, ρ_{uv} corresponds to h_m if $c(u) = c(v)$, otherwise $\rho_{uv} = (1 - h_m)$. Considering the in-process homophily values for the minority and the majority group, respectively expressed as ρ_m and ρ_M, these two parameters are directly proportional to the observed homophily indicated as h_m and h_M. In general, fixing $\rho = 0.5$ for one class generates a neutral group ($h = 0$), $\rho > 0.5$ generates a homophilic group ($h > 0$) and, finally, $\rho < 0.5$ generates a heterophilic group ($h < 0$). The process designed to generate a bi-populated graph $G = (V, E, c)$ is the following:

1. **Initialization.** $|V| = N$ is the network size and d_{out} the number of outgoing out-links from each new node (i.e., $|E| = N \times d_{out}$). Then d_{out} nodes are initialized, forming a fully-connected graph. To reduce randomness, in the initialization phase there is no real majority class, since the two groups are equally distributed.

2. **Add node.** A new node v is added to the graph, belonging to the minority with probability p_m.

3. **Add edges.** For the new node v, we generate d_{out} out-links, each one with the following probability that incorporates both in-process homophily and rich-get-richer effect:

$$p_u = \frac{\rho_{vu}\big(d^{in}(u) + A\big)}{\sum_{w \in V} \rho_{vw}\big(d^{in}(w) + A\big)}$$

   The A constant, introduced in the original Biased Preferential Attachment model to avoid penalizing new nodes, is fixed to 1.

The process terminates when the graph reaches $|V| = N$.

4.5.2 Impact of Homophily

In this first set of experiments, we aim at investigating homophilic and heterophilic situations for both groups. We keep the same minority/majority

partition ($s_m = 0.3$) with networks having 10,000 nodes each and, to show more robust results, each configuration expressed in terms of (ρ_m, ρ_M), is tested 10 times. Consequently, metrics computed over networks with the same configuration are evaluated through their average. We generate three distinct groups of configurations:

- **S1.** We create a neutral majority with $\rho_M = 0.5$, and vary the level of homophily of the minority $\rho_m \in [0.2, 0.9]$.
- **S2.** We create a neutral minority with $\rho_m = 0.5$, and vary the level of homophily of the majority $\rho_M \in [0.2, 0.9]$.
- **S3.** We create a homophilic majority and a homophilic minority, testing 4 possible configurations of (ρ_M, ρ_m) in the set $\{0.7, 0.9\}$.

Figure 4.6 presents the overall exposure, $\Delta(\mathcal{V})$, given by the different recommenders, comparing the two settings in which a group is homophilic but the other is not (S1 and S2). Looking at the $\Delta(\mathcal{V})$ obtained in configuration S1 (left side in Figure 4.6), the minority indeeds obtains more exposure when it is homophilic. In particular, the more the minority is homophilic, the more exposure it gets. In contrast, if the minority is heterophilic, it is the majority that benefits in terms of exposure. Although all the recommenders behave similarly, these effects are more evident in ALS and SLS. In S2, the homophilic majority leads to an analogous effect; indeed when homophilic, it receives more exposure, while when heterophilic, it facilitates the neutral minority to get more than their representation (right side of Figure 4.6).

The analysis of the overall exposure in the case in which both groups are homophilic (S3) is presented in the heatmap in Figure 4.7. The x-axis represents ρ_M, while the y-axis reports the ρ_m values; each cell of the matrix contains the values of $\Delta(\mathcal{V})$ under that setting. In case of neutral homophily for both groups ($\rho = 0.5$), no disparate exposure is given by any of the algorithms (except for ADA, who gives a slight advantage to the minority). For the scenario in which both groups are highly homophilic ($\rho = 0.9$ and $\rho = 0.7$), the majority is slightly advantaged. The

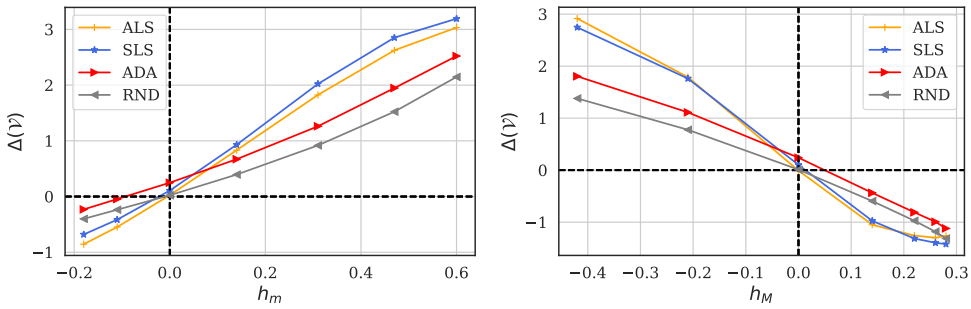


Figure 4.6: (Best seen in color.) Distribution of $\Delta(\mathcal{V})$ observed in S1 and S2. The minority comprises 30% of the nodes ($s_m = 0.3$). In the left plot, the majority is neutral and the heterophily/homophily of the minority varies. In the right plot, the minority is neutral and the heterophily/homophily of the majority varies.

worst scenarios can be observed in cases of one extremely homophilic class and neutral the other (top left and bottom right cells of each matrix), which present the values of $\Delta(\mathcal{V})$ with strongest intensity in absolute value (again, this phenomenon is especially emphasized by ALS and SLS). These extremes cases capture a trend in each heatmap in the figure, which indicates that as soon as a group increases its homophily, it increases its exposure.

To further confirm the role of homophily when considering exposure received at individual level, as previously investigated for real data, we focus on the ranking generated either by exposure ψ or by degree-normalized exposure ψ/d_{in}. We look at the fraction of nodes belonging to the homophilic class in the top-10% and top-20% of the most recommended nodes. Since we are capturing the "rich-get-richer" and "individual fairness" phenomena we previously captured for the minority group, in Figure 4.8 we report the results for S1. The first row of the figure shows that a stronger homophilic tendency leads to present the homophilic class in the highest positions of the recommendations, for all the algorithms. While, looking at the second row, where nodes are ordered by exposure

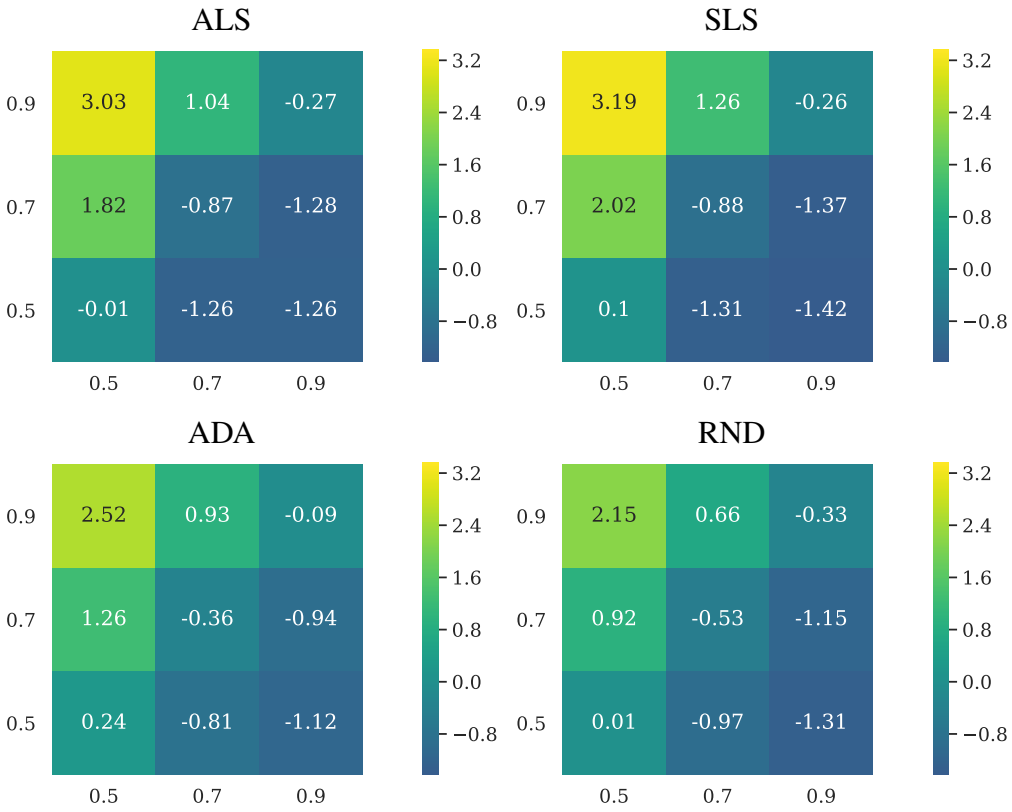


Figure 4.7: (Best seen in color.) Exposure $\Delta(\mathcal{V})$ computed over networks characterized by different homophily of the minority ρ_m (y-axis) and homophily of the majority ρ_M (x-axis).

normalised by the in-degree, the effect is mitigated. In particular, for ADA, the configuration with small homophily presents a minority still underrepresented.

4.5.3 Impact of Minority Size

Synthetic networks also enable us to investigate how exposure varies with the relative size of the minority in the graph. To do so, we generate

a fourth group of configurations, **S4**. Keeping the minority homophilic ($\rho_m = 0.8$) and the majority neutral ($\rho_m = 0.5$), we range the minority size s_m from 5% to 45%.

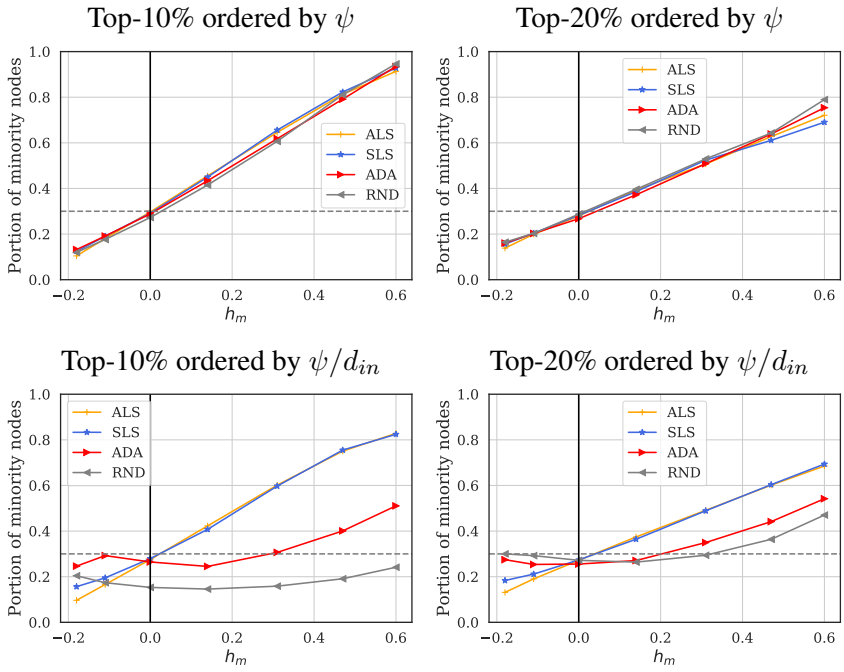


Figure 4.8: Fraction of minority class in S1 in the top positions of rankings ordered by exposure ψ (first row) and by degree-normalized exposure ψ/d_{in} (second row).

Each configuration in S4, characterized by a different s_m, corresponds to a graph with 10,000 nodes and is generated 10 times (again, the results we present are an average of those obtained for the 10 networks depicting the same configuration). The observed homophily (h_m) presents $\mu = 0.4$ and $\sigma^2 = 0.01$, showing that the data generation process is stable with respect to the different h_m.

In Figure 4.9, we report the $\Delta(\mathcal{V})$ obtained for each configuration. We observe that being a small minority ($s_m = 0.05$) can mitigate the ho-

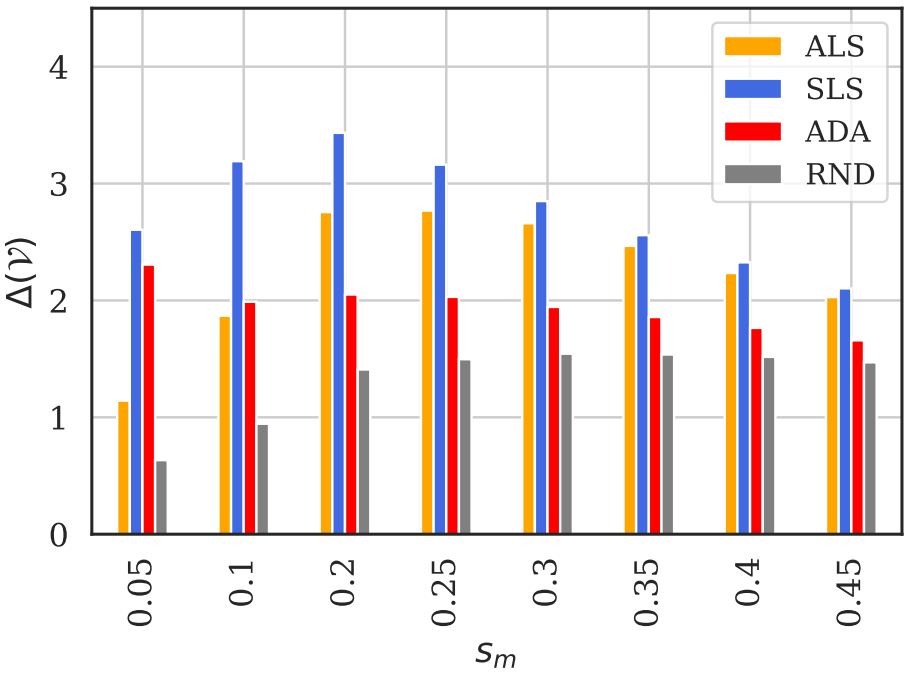


Figure 4.9: Distribution of $\Delta(\mathcal{V})$ for different minority sizes s_m and a homophilic minority ($\rho_m = 0.8$). The size of the minority does not have an effect on exposure as dramatic as the effect of homophily.

mophily effect, while keeping the minority with a size much lower than half of the graph ($s_m \in 0.1, 0.2, 0.25$) can positively impact the final gain in exposure. Despite these considerations, the size does not impact exposure as much as homophily, since $\Delta(\mathcal{V})$, for each recommender, ranges in a small interval.

Observation 6. *The relative size of the minority does not impact exposure as much as homophily does.*

4.6 Summary

In this Chapter we analyze through an empirical study the exposure of minorities generated by the People Recommender Systems and how they are impacted by the level of homophily in the social graph. The fact that homophily plays a key role in the exposure that is given to a group, sometimes regardless to the fact that a group may be a minority in the network, is the main take-home message of this work. Also, the rich-get-richer phenomenon stimulated by the algorithmic suggestions, appears to be stronger within the minority class compared to the majority class, especially when the minority is homophilic. Normalizing our findings through the in-degree, considered as measure of initial social capital, we show that nodes in the minority class are constantly disadvantaged in terms of exposure. Finally, we confirm our findings in real-world networks proposing a new synthetic graph generator, which allows us to extend the analysis to a wide range of configurations.

5

Exposure Inequality in People Recommender Systems: The Long-Term Effects

5.1 Introduction

In the previous chapter we highlighted the harmful consequences of link recommenders after one round of recommendations [FBBC20]. Such a static picture can be limited as it does not study the consequential effects of the user behaviour which, by accepting or rejecting the recommendations, can determine the future structure of the social network and thus the exposure distribution. Specifically, multiple interactions between users and recommendation algorithms tend to nourish a *feedback loop*: i.e., the output generated by the recommendation algorithm is then fed as future input for the next training of the recommender. In our setting, the recommended new links which are accepted, modify the structure of the network, thus constituting the input for the next cycle of link suggestions. In the context of items recommendation, recent simulation-based studies interested in the side-effects of collaborative filtering algorithms, show how a similar feedback loop [MAP+20] impacts over user preferences, stimulating the *popularity bias* [YHT+21]. Those works underline the

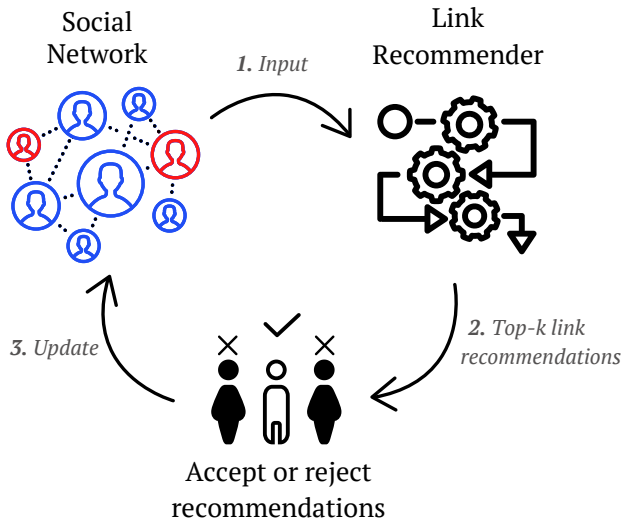


Figure 5.1: Bird's eye view of the simulation framework.

importance of providing models able to simulate the potential scenarios which may be otherwise difficult to investigate.

Following a similar approach, but focusing on people recommendations, in this chapter we tackle the following research question: *"which impact can link recommendation algorithms have over the network structure and user exposure along multiple rounds of recommendations?"*

Our contribution towards answering this question is a model able to simulate the long-term consequences of the injection of new recommended links into the network, reproducing the feedback loop triggered by the multiple interactions between users and the link recommender. Figure 5.1 provides a bird's eye view of our proposed simulation model.

Starting with a social network where the nodes are partitioned in subgroups, e.g., by means of protected attributes such as gender or race (§5.3.1), a set of different link recommenders are applied to the network to provide, at each iteration, k link recommendations to each user (§5.3.2). The user at this stage may then decide to accept or reject the recommendation. This decision is governed by three different stochastic user behavior models (§5.3.3). The rejected links are then discarded, while the

new ones are included in the social graph. The new augmented graph will then be the input to the next round of training of the recommender. Despite our models does not consider the organic growth of the social graph, the simulations show that the injection of new links proposed by the recommendation algorithms can move the social graph far from the initial configuration.

Contributions and findings. In this chapter we propose a simulation model able to utilize several network configurations, user behaviors, and recommendation models in order to study the long-term effects of people-recommender systems in social networks. We quantify the long-term disparate exposure generated by different initial network topologies, minority size and homophily level, and using different state-of-the-art link recommenders, and different stochastic user behavior models. Our work confirms and extends the preliminary theoretical insights provided by [SRC18] and the empirical results of our previous chapter [FBBC20], which was limited to one single round of recommendations.

Our findings are summarised as follows:

- Confirming the theoretical findings of a recent work by [SRC18], our experiments show that, if the minority class is homophilic enough, it can get an advantage in exposure from all link recommenders. If the minority is heterophilic instead, it gets underexposed.
- While the previous observation is robust to all the recommenders, the speed and magnitude of the disparate exposure along time differ across recommenders.
- While the homophily of the minority affects the speed of the growth of disparate exposure, the size of the minority affects its magnitude.
- The user behavior model (how recommendations are accepted or declined) does not impact significantly the evolution of exposure as much as the initial network configuration and the algorithm do.
- Some recommenders can strengthen exposure inequalities at the individual level: after a few iterations, most of the links are recom-

mended towards a small subset of "super-star" nodes. This happens for both the minority and the majority class and independently of their level of homophily. Hence, in the long-term, the "rich-get-richer" effect is exacerbated .

In the rest of this chapter, we first review the literature relevant to our work in Section 5.2. Then, we introduce the simulation model in Section 5.3. Our results are presented in Section 5.4. Finally, we conclude this chapter in Section 5.5.

5.2 Related Work

In this section we discuss the literature most related to our work. We divide the presentation into two topics: work dealing with inequalities in social networks, and simulation-based studies in recommender systems.

Inequalities in social networks. In the previous chapter we observed, in a "static" single round of recommendations, that homophily is a driving force in shifting visibility distribution. In particular, we introduced the concept of disparate visibility in a bi-populated network, showing how effects such bias in rankings and rich-get-richer can get amplified by homophilic networks [FBBC20]. The main limitation of our previous study is that it looks at one single round of recommendations, missing the long-term effects.

A recent work by [LKW+19] shows that the perceptions about the size of minority groups in social networks can be biased, often exhibiting systematic over- or underestimation. Moreover, these biases can be explained by the level of homophily and by the size of the minority class. Our work, is inspired by their insights, extending the analysis of the inequalities while the network is injected with new links driven by the recommendation output. We confirm their observations, showing how the recommender algorithms can introduce even more inequality along the time.

A pivotal contibution proposes methods for fairness-aware link analysis, introducing techniques able to mitigate unfairness generated by Pager-

ank [TPT+20]. Later, in §5.4, we will show that another popular random-walk based recommender (i.e., SALSA) can increase the unfairness in visibility in the long run, thus confirming the need to devise methods able to mitigate these effects.

Simulation-based studies in recommender systems. In a first work, [CMMB21] combine link recommendation and opinion-dynamics models in a simulation-based framework, to assess the effect of people recommenders on the evolution of opinions in a social network. They show that, if the initial network exhibits high level of homophily, people recommenders can help creating echo chambers and polarization.

In the context of collaborative-filtering-based methods, it has been shown that popularity bias can be stimulated by feedback loop, where popular items tend to obtain more and more interactions if generated through recommendations [MAP+20]. In the same direction, a theoretical framework has been proposed to model the effects of "*filter bubble*", i.e., the tendency of the recommendation algorithm to drive the preferences of the user towards a limited amount of items. [JCL+19]. Similarly, a work by [YHT+21] proposes a simulation model for measuring the impact of recommender systems over time, analyzing the changes in the user experience with an application designed for food recommender system.

Our work is motivated by the importance of studying algorithmic bias in recommendations and rankings in the long term, i.e., beyond the single round of algorithmic intervention. In this regard, [GLG+21] have recently introduced the problem of *long-term fairness*, designing also solutions able to account for algorithmic unfairness in the long-term in movies recommendations. Moreover, [SSL+17] propose a simulation model able to include multiple recommender systems combined with different users choice models, proving that the rich-get-richer effect tends to increase over time, stimulated by the algorithm. In our study we analyze the evolution of rich-get-richer effect in social networks, fueled by the edges created thanks to the recommendation algorithms.

The feedback-loop effect has been audited by [NHH+14], showing how in the case of MovieLens data, recommendations generated through a collaborative filtering approach have not strengthen the filter bubble

effect. In our work we go in the opposite direction, showing how homophily may generate biased recommendations, towards a smaller set of recommendations, reducing the diversity of those. In the same direction, a recent work proposes a mitigation strategy to reduce popularity bias in recommendations through different methods based on active-learning [SKNS19]. The methods proposed are aimed at reducing popularity bias, which in our setting can be related to rich-get-richer effect. Although this work may be used to propose mitigation strategies to reduce inequality in exposure, the main weakness of their work regards the lack of analysis of different input data distribution. This kind of analysis may help to understand which kind of distribution of user-item interactions may benefit more from their method.

5.3 Model

We consider a social graph whose nodes are partitioned by demographics (e.g. gender, age or other characteristics). More formally, let $G = (V, E, \ell)$ be the social graph, where V is the set of nodes, $E \subseteq V \times V$ is the set of directed edges, such that an edge $(u, v) \in E$ indicates the fact that u *follows* v. Finally $\ell : V \to \{V_m, V_M\}$ is a labeling function assigning each node to either the minority (V_m) or the majority (V_M) class (with $|V_m| < |V_M|$). We denote by $s_m = |V_m|/|V|$ the fraction of nodes belonging to the class less represented in the network, i.e., the *minority*, and by s_M the fraction of nodes belonging to the majority

Homophily. As for the previous chapter, to capture the bias in the distribution of the edges towards each group, we introduce a measure of homophily, expressed as *the tendency of people in a a group to connect to individuals in the same group*. We model the homophily as the portion of edges distributed within the same group discounted by the fraction observed in a random configuration [FBBC20]. More formally:

$$h_i = \frac{|E_{ii}|}{|E_{i.}|} - s_i \tag{5.1}$$

where $E_{ii} = \{(u, v) \in E | u \in V_i \wedge v \in V_i\}$ and $E_{i\cdot} = \{(u, v) \in E | u \in V_i\}$. This measure ranges in the interval $(-s_i, 1-s_i]$. A group is called *homophilic* if the tendency to connect to nodes of the same group is stronger than expected ($h_i > 0$), *heterophilic* when this tendency is weaker than expected ($h_i < 0$), and *neutral* if the number of edges within the group is comparable to the relative size of the group ($h_i = 0$).

Step-by-step. In our simulations we reproduce the multiple interactions between the users and the recommendation algorithm, where at each round, a set of new links is recommended to a portion of users randomly sampled from the graph. This sampling represents the fact that only a set of users are online at a certain time, helping reproducing a more realistic scenario. Then the users accepts or rejects the recommendations according to a given stochastic user behavior model. This process is iterated for a given number of iterations.The graph grows accordingly to the new accepted recommendations: neither organic growth, nor edge removal are considered. Table 5.1 summarizes the simulation process step-by step.

For all the results that we report in §5.4 we use $T = 20$, $\alpha = 20\%$ and $k = 3$.

We next present in more details the various key components of our model.

5.3.1 Initial Network Configuration

In order to control the level of homophily and the size of the minority class, while keeping a realistic network structure, we propose a novel data generation process which, starting with a real-world bi-populated network, performs just the minimum amount of node class-swappings and link rewirings to match the requested levels of homophily and size of the minority class. In this regard, our networks are *semi-synthetic*. The process is explained in details next.

Our starting real-world network comes from Tuenti, a social network popular few years ago in Spain, which was known as "the Spanish Facebook". The dataset includes demographic information about users as gender and age [LVKK16]. The network has 8,983,560 nodes (users) and

Table 5.1: Simulation steps.

1. **Input.** We start with an initial network configuration with specified levels of homophily and size of the minority class (how this initial configuration is generated by modifying a real-world social graph is presented in §5.3.1). We also set parameters such as the number of recommendations k that a user receives in a round, the number of iterations T, the fraction α of users to sample, the link-recommendation algorithm A (presented in §5.3.2), and the stochastic user behavior model B (discussed in §5.3.3).

2. **Recommendation round.** A link recommender model is trained over the current social graph by the algorithm A. A portion α of users is sampled from the network. Those sampled users receive their top-k recommendations each. The recommendations are links never recommended before and are generated from the set of missing edges at distance two (e.g. "friends of friends").

3. **Graph update.** Each user decide to accept or reject each of the k recommended links, according to the model B. The social graph is thus updated by adding the newly accepted links. Each link rejected at this stage is discarded and never recommended again.

4. **Repeat.** Steps 2 and 3 are repeated T times.

17,830,103 edges, where a generic edge (u, v) indicates a user posting on another users' *wall*. Along the chapter, we use a sample of the original network used in our previous work [FBBC20]. This sample, which is also the one from which we derive the other configurations, is given by partitioning the users by age (16 as cut-off). We call it **G0**: it contains 500,000 nodes and 2,813,744 edges, with a relevant minority ($s_m = 0.30$) and both groups (majority and minority) appearing to have some level of homophily. Starting from this network, we generate 4 different semi-synthetic networks, ranging different values of h and s. More in details,

let V_m and V_M be respectively the set of minority's and majority's node in the input network and $N_m = |V_m|, N_M = |V_M|$. Let also h_m, h_M indicate the actual homophily of each class, and finally s_m and s_M are the relative size of both classes. The generation process takes as input the desired level of homophily for both classes, denoted h^*_m, h^*_M and the desired proportion for the minority class s^*_m, and works as follows:

1. **Change minority-majority size.** Let $N^*_m = (N_m + N_M)s^*_m$. If $N^*_m < N_m$, then $N_m - N^*_m$ nodes are sampled at random from the minority class V_m and their label is flipped to the majority. Otherwise, we sample of $N^*_m - N_m$ nodes are extracted from the majority V_M and their label flipped to the minority.

2. **Change homophily.** For each group i $\in \{m, M\}$ we first compute the difference between the initial and the final homophily $|h^*_i - h_i| = B_i$. Then depending on the sign of the difference $h^*_i - h_i$ we define which edges need to be rewired. Rewiring an edge (u, v) means substituting (u, v) with an edge (u, w) such that $\ell(v) \neq \ell(w)$. If $h^*_i - h_i > 0$ a sample of edges beloning to $E_{ij} = \{(u, v) \in E | u \in V_i \wedge v \in V_j\}$ is selected and rewired towards nodes in V_i. In this way we increase the set of nodes in E_{ii}, reaching the requested level of homophily. Viceversa, if $h^*_i - h_i > 0$ the operation is the opposite: old edges belonging to the subset E_{ii} are rewired towards nodes in V_j. In both cases, the final amount of edges rewired is $E_{i.} \times B_i$.

Since some recent literature has shown that small subpopulations within a social network can impact the whole graph [SRC18, FBBC20, KGW$^+$18], we generate networks with biased distributions for the minorities. Only for one case, to have a comprehensive analysis, we modify the homophily level in both minority and majority groups.

More in details, these are the configurations we focus on:

- **G1.** To analyze the effect of a **small homophilic minority** we generate a graph with $s_m = 0.1$ and $h_m = 0.4$, with a neutral majority.

Table 5.2: Table summarizing information about the generated graphs. For each one we have: i) name, ii) scenario characterizing the network

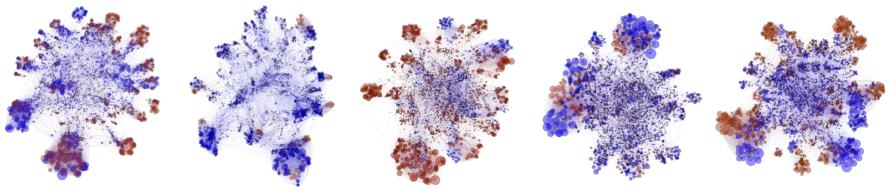
Graph	Scenario	s_m	h_m
G0	*original*	0.3	0.42
G1	*different sizes + homophilic minority*	0.1	0.4
G2	*same sizes + homophilic minority*	0.45	0.5
G3	*different sizes + heterophilic minority*	0.3	-0.25
G4	*different sizes + homophilic groups*	0.3	0.6

- **G2.** To emphasize the role of homophily we also generate a graph with **comparable sizes** between the two groups ($s_m = 0.45$) with the minority strongly homophilic ($h_m = 0.5$) and a majority still neutral.
- **G3.** This configuration is the unique with a **small heterophilic minority** ($h_m = -0.25$) and a neutral majority.
- **G4.** The the final configuration has **both groups homophilic**. In particular, we keep a small minority ($s_m = 0.3$) and both groups with high level of homophily ($h_m = 0.6$ and $h_M = 0.2$).

$G1$ and $G2$ are a useful comparison against $G0$, since they present comparable level of homophily but different sizes of the minority, while $G3$ is useful to explore the heterophilic case and $G4$ resembles a scenario quite common in contexts where phenomena such as polarization and filter bubbles drive the network formation [GMGM18]. Table 5.2 summarizes the five networks used in our analysis, while Figure 5.2 depicts a sample of each network.

5.3.2 Link Recommenders

Link recommendation algorithms are selected accordingly to state-of-the-art performance and popularity in the literature [LFS17a, FBBC20, SRC18]. Specifically, we select the same subset of algorithms used in the



(a) G0 (original) (b) G1 (homophilic minority) (c) G2 (one homophilic group) (c) G3 (heterophilic minority) (d) G4 (two homophilic groups)

Figure 5.2: Representation of a sample of each generated network, where the minority is indicated in red, while the majority in blue. Each sample considers 5,000 nodes and those are the ones with highest degree in each group. Specifically for each group, a total of $s_i \times$ 5,000 ($i \in \{m, M\}$) nodes are sampled.

previous chapter: (i) Adamic-Adar (**ADA**), (ii) Stochastic Approach for Link-Structure Analysis **SLS**, (iii) Alternating Least Squares **ALS** and (iv) a random baseline **RND**.

5.3.3 User Behavior Models

In order to simulate the user feedback on the received recommendation, we consider three stochastic user behavior models. The first two are adapted from a recent work simulating user-item interactions [YHT+21], while the third one defines acceptance probability biased by the position in the ranked list of recommendations. Through these stochastic choice models the users add in expectation one edge per recommended list.

- **B-LZY** - *Lazy*. The user accepts directly the first recommendation:

$$P(u \textit{ selects } v \textit{ at position } i) = \begin{cases} 1 & \text{if i = 1} \\ 0 & \text{otherwise} \end{cases}$$

- **B-RND** - *Random*. The user picks in the top-k, one recommendation uniformly at random:

$$P(\textit{u selects v at position i}) = \frac{1}{k}$$

- **B-PSB** - *Position Biased*. This policy refers to the idea of having user choices biased by the position bias of rankings, where the user may accept or reject the recommendation with probability based on its position [CZTR08]. Hence, higher ranked suggestions are more likely to be chosen:

$$P(\textit{u selects v at position i}) = \frac{1/\log(i+1)}{\sum_{j=1}^{k} 1/\log(j+1)}.$$

- **B-MIX** - *Mixed*. In order to evaluate how heterogenous user behaviors may affect the exposure distribution, we also include B-MIX, which is a combination of the previous policies. Specifically, at each iteration, each user first picks, uniformly at random, one of the three strategies above, then follows it.

B-PSB model resembles the classical position bias, observed as key factor for predicting clickthrough rates in search engines [CZTR08, Joa02]. The other two user behavior resembles two extreme situations: i) B-RND is the one less dependent by the order of the recommendation list; while B-LZY represents the case in which the user relies completely on the order imposed by the recommendation algorithm.

5.4 Results

In this section we present the results of our experiments, focusing on the key measure that we call *exposure* of the minority, which is simply defined as *the portion of total number of recommendations which suggest a node of the minority*, and denoted $\mathcal{E}_m$. Note that the total number of recommendations is constant and corresponds to $k|V^{\alpha}|$.

5.4.1 Exposure in the Long-run

Figure 5.3 shows the trend of the exposure of the minority for each of the four networks (G0 is omitted for space limitation as it presents results almost indistinguishable from G4). For each experiment, we track the exposure and the percentage of new edges added at each iteration with respect the original network. The dashed line represents in each plot the relative size of the minority and the user choice model is fixed to B-PSB. In cases when the minority is homophilic (G1, G2 and G4), generating recommendations through ALS and SALSA leads to a positive trend of growth for the exposure of the minority. For the other two recommenders (ADA and RND) the effects described above are still present but less visible.

For the case in which the minority is heterophilic (G3) the exposure decreases weakly, slightly benefitting the majority. This is the only case when the exposure distributed to the minority is less than its relative size. It is also evident how the collaborative filtering approach (ALS) and the random walk based model (SLS) contribute more to reduce the exposure allocated to the minority with respect the other two models (ADA and RND). For all the networks characterized by an homophilic minority (G1, G2 and G4), the growth in the case of the collaborative filtering approach (ALS) is faster in the first steps and then stabilizes to a steady-state to the rest of iterations. While, for the random walk solution (SLS), the trend starts at similar values of exposure, but then grows constantly.

Observation 7. *The disparate exposure grows after each iteration in favour of the minority, when it is homophilic. On the other hand, an heterophilic behavior of the minority does not impact abruptly its exposure. When both groups are homophilic, the recommender still increases the exposure of the minority. The severity of all those effects is stronger when using ALS and SLS and weaker for ADA and RND.*

To further investigate the differences in growth of exposure, we track $\mathcal{E}_t/\mathcal{E}_1$ for $t \in \{2, ..., T\}$, which is relative quantity of exposure mea-

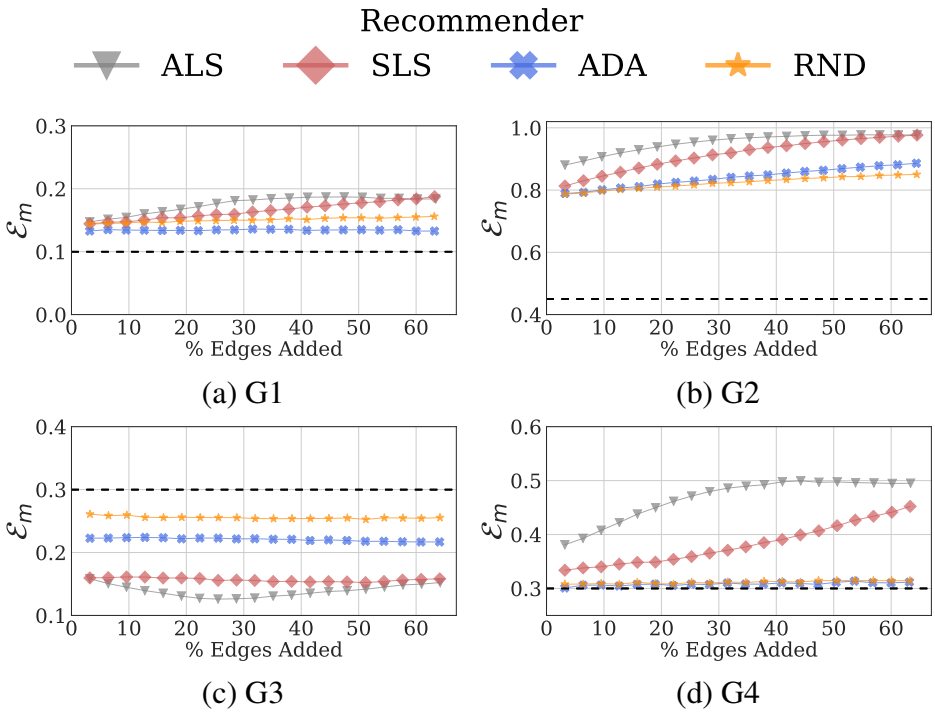


Figure 5.3: Exposure of the minority ($\mathcal{E}_m$) along time, for different recommenders and one fixed user behavior (B-PSB).

sured with the respect to the first iteration. In order to analyze the transition phases previously mentioned, we focus respectively on the iterations $t \in \{2, 10, 20\}$. In Figure 5.5 we plot those values on the y-axis and the iterations on the x-axis. As suggested by the previous plots, with the ALS recommender, when the minority is homophilic its exposure tends to increase faster in the first iterations to then stabilize. On the other hand, SLS presents a continuous increase, without slowing down the process after 20 iterations. The stronger growth comes from cases where the differences in sizes between minority and majority is relevant ($s_m = 0.3$) and the minorities are homophilic (G0, G1 and G4). This means that, even when also the majority is homophilic, the effect is still present, showing again

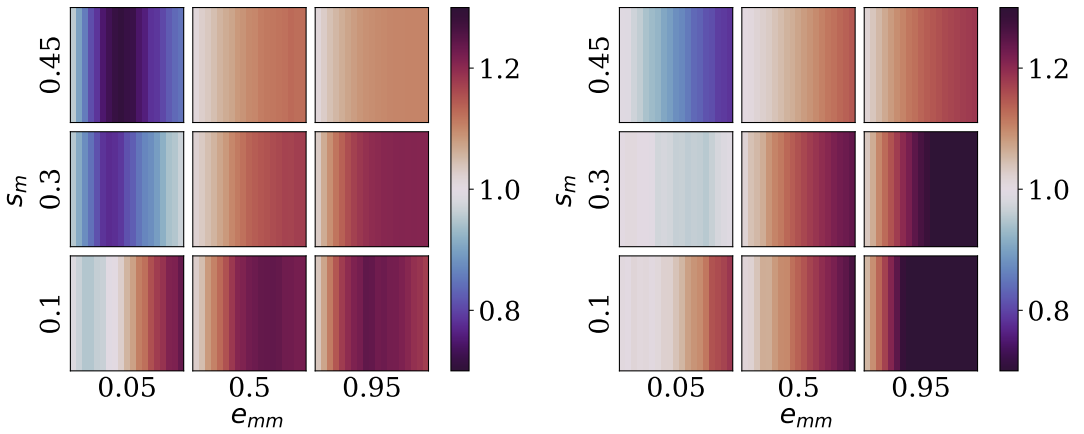


Figure 5.4: Heatmaps describing the evolution of $\mathcal{E}_t/\mathcal{E}_1$ in $T = 20$ iterations, computed over 9 configurations which are small variants of G1, G2 and G3: $e_{mm} \in \{0.05, 0.5, 0.95\}$ (x-axis) and $s_m \in \{0.1, 0.3, 0.45\}$ (y-axis), all having neutral majority $h_M = 0$. ALS recommender (left-hand side), SLS recommender (right-hand side).

that having both groups homophilic does not imply a benefit for the majority. As already seen in Figure 5.3 and as expected, ADA and RND do not produce much exposure disparity, even a slight advantage for the minority class can be observed for the cases in which the minority is homophilic.

Observation 8. *Different recommenders exhibit different influence on exposure along time. ALS increases exposure inequality in the first iterations, then stabilizing in a steady state. SLS instead keeps increasing disparate exposure constantly.*

We further extend this analysis, exploring a wider range of initial configurations, with the aim of disentangling the effects of size and homophily along time. For this purpose, as the size of the minority s_m is part

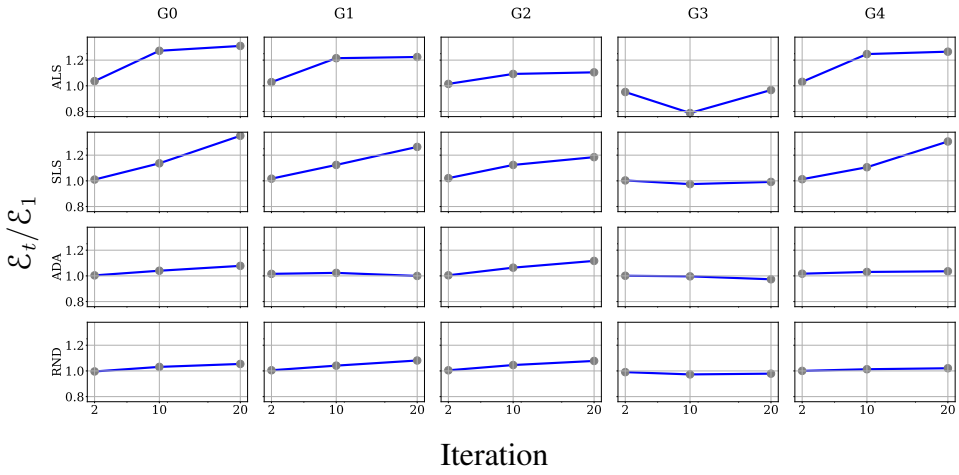


Figure 5.5: Evolution of exposure relative to the one observed at first iteration $\mathcal{E}_t/\mathcal{E}_1$, after 1, 10 and 20 iterations.

of the definition of the homophily h_m (Eq. 1), we use directly the fraction of edges which, starting from the minority, remain in the minority: i.e.,

$$e_{mm} = \frac{|E_{mm}|}{|E_{m.}|}$$

In particular, we produce 9 configurations which are small variants of G1, G2 and G3: $e_{mm} \in \{0.05, 0.5, 0.95\}$ and $s_m \in \{0.1, 0.3, 0.45\}$ (all having neutral majority $h_M = 0$).

Each box of the heatmaps in Figure 5.4 represents, for one configuration, the evolution of $\mathcal{E}_t/\mathcal{E}_1$ along $T = 20$ iterations.

Analyzing the two heatmaps, comparing the boxes by columns and posing the attention on a single row, we observe that both effects already observed in the previous experiments, i.e. the steady-state generated by ALS and the constant growth caused by SLS, change in terms of severity but not in timing. This means that the variation of $\mathcal{E}_t/\mathcal{E}_1$ can be less or more severe, depending on the distribution of e_{mm}, but the pace to which process evolves is the same. Analyzing the heatmaps by rows, and posing the attention on a single column, is evident how the size of the minority

(represented here by s_{mm}) can influence the pace of the effects but not the range of values (color intensity) of $\mathcal{E}_t/\mathcal{E}_1$.

Observation 9. *The homophily of minority can impact the speed at which the growth of exposure disparity occurs. On the other hand, the severity of this effect is mostly determined by the size of the minority.*

5.4.2 Effect of User Behavior Models

In all the experiments presented so far we were adopting the B-PSB (position bias) user behaviour model. We next analize the effect of different user behaviour models. Figure 5.6 reports the exposure of the minority tracked under three different policies on G0. Each plot represents a recommender and each line in the plots represents the trend of $\mathcal{E}_m$ for one user behavior. For all the plots there is not such a significant difference in trends between models. This means that in circumstances where user behavior is either homogenous (B-LZY, B-PSB and B-RND) or heterogeneous (B-MIX), and the organic growth of the network is not considered, the effect of the recommenders are consistent.

Observation 10. *The different user behaviour models do not impact the exposure in our simulations as much as the type of recommender system and the initial configuration of the network do.*

5.4.3 Rich-get-richer Effect

After having analyzed the inequality in exposure at the group level, we now focus on the in-degree distribution at the individual level, focusing on the relationship with the popularity of the nodes (number of followers or in-degree).

As already observed in the literature, new links injected in the network can alter the inequality in the distribution of in-degree [SSG16]. For this

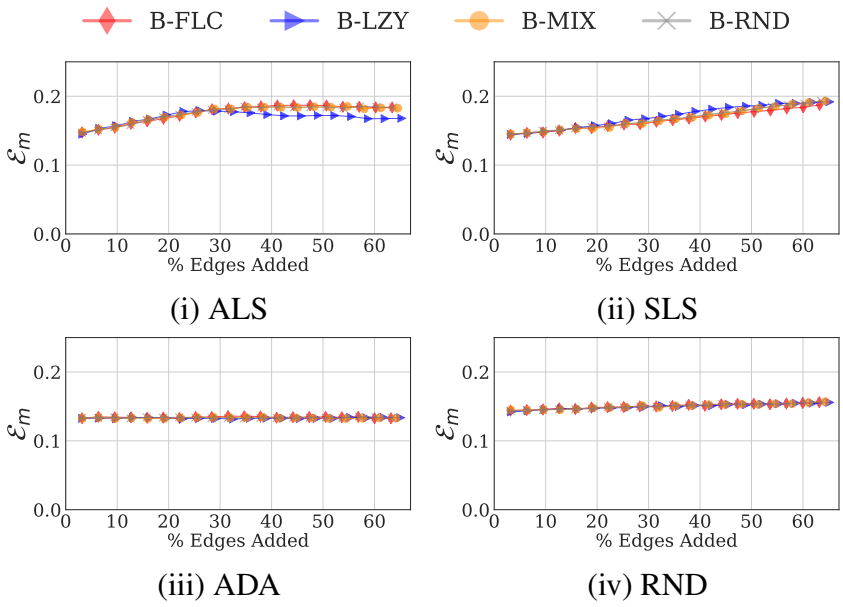


Figure 5.6: Exposure of minority when using different acceptance policies, running on G0.

reason, we study here the evolution of the rich-get-richer effect within the two groups, focusing the attention on how the in-degree of nodes is altered by the recommended output in the long-term. We compute the Gini coefficient to analyze the level of concentration of in-degree within the two group of nodes, after each iteration. Although it was introduced in economics to measure the income or wealth inequality, Gini coefficient is widely used to measure inequalities in general [HL10]. It is defined as follows:

$$G = \frac{1}{N}\left(N + 1 - 2\frac{\sum_{i=1}^{N}(N+1-i)y_i}{\sum_{i=1}^{N} y_i}\right).$$

In our context N is the number of nodes and y_i is the in-degree of the *i-th* node, which has been indexed in ascending order by $y_i \leq y_{i+1}$. The index ranges from zero to one, where if all nodes receive the same amount

of quantity (in-degree) then it is 0, and 1 if only one node receive the total amount. Thus, the higher the coefficient is, the higher the inequality distribution is as well. Figure 5.7 reports the Gini index in the long-run. Each row indicate the network, each column of plots refers to a group (minority or majority) and each line shows the Gini index after each iteration, when using all the four recommenders and one user behavior $(B - PSB)$.

For all the networks we can observe a rich-get-richer effect in both the minority and the majority class: inequality of in-degree, as expressed by the Gini index, keeps growing, meaning that the high-degree nodes keep receiving more and more recommendations. Among the recommenders, ALS and SLS present a faster growth of the in-degree inequality, while ADA and RND are by far slower. When the two groups have comparable size and only one of them is homophilic (G2), the non-homophilic group gets a less severe effect. It is also evident that when the minority is heterophilic, the more impacted group is the majority, which even if not presenting biased preferences (either homophilic or heterophilic), experiences a stronger positive trend for the growth of Gini index.

The homophily level of one group impacts, not only the inequality in in-degree distribution inside the group, but indirectly it affects also the inequality in the rest of the graph.

After having observed an exacerbation of the rich-get-richer effect in the long-run, we next monitor the distribution of exposure between nodes in the two groups. We study how subset of nodes, grouped by different in-degree, can be exposed differently in the long-run. In Figure 5.8 we show the cumulative distribution of exposure accumulated by nodes ordered by their in-degree. In particular, each bar is divided by colors, where starting from the bottom, it represents the subset of nodes having at most the correspondent in-degree. This means that, for example, the first three colors (from the bottom) represents the first 5% of the nodes having the highest in-degree. On the y-axis we track the fraction of visibility accumulated by the nodes.

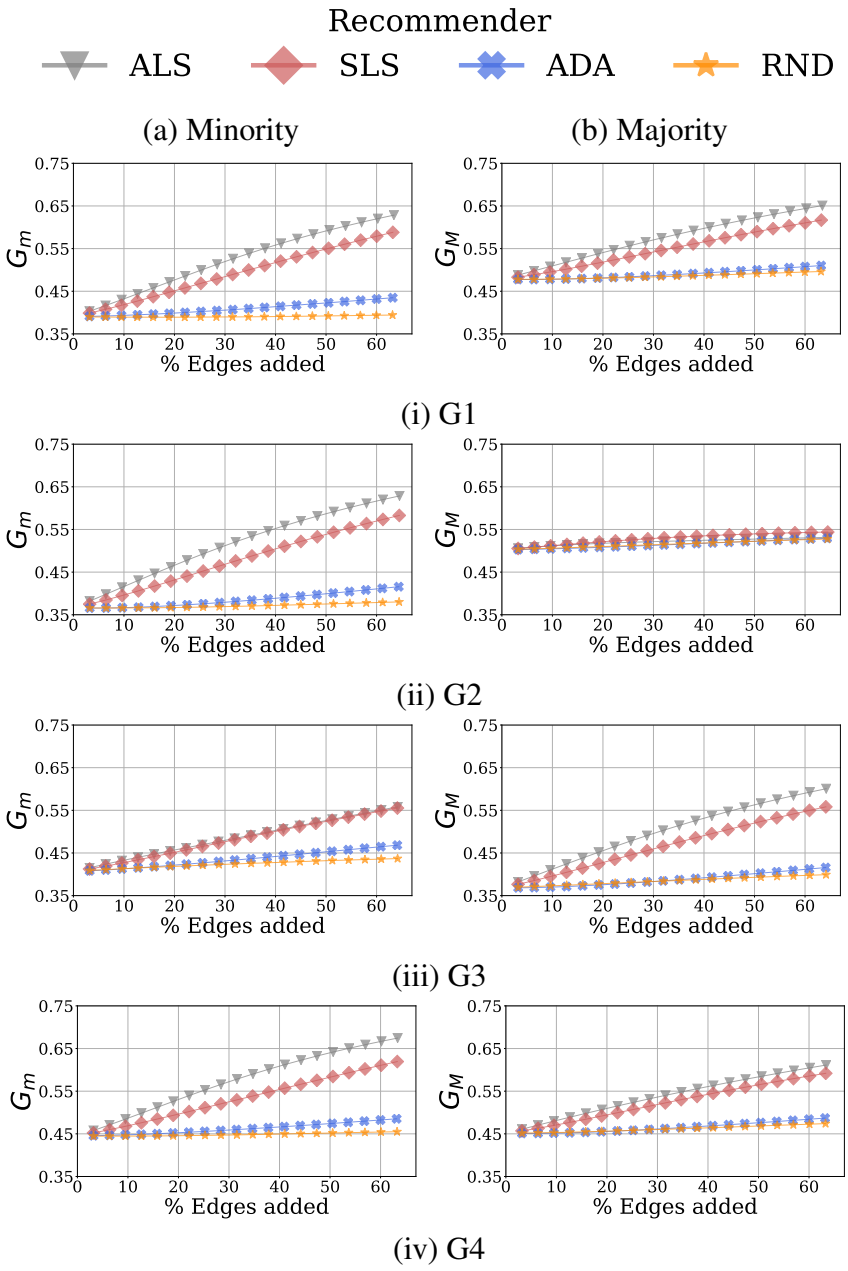


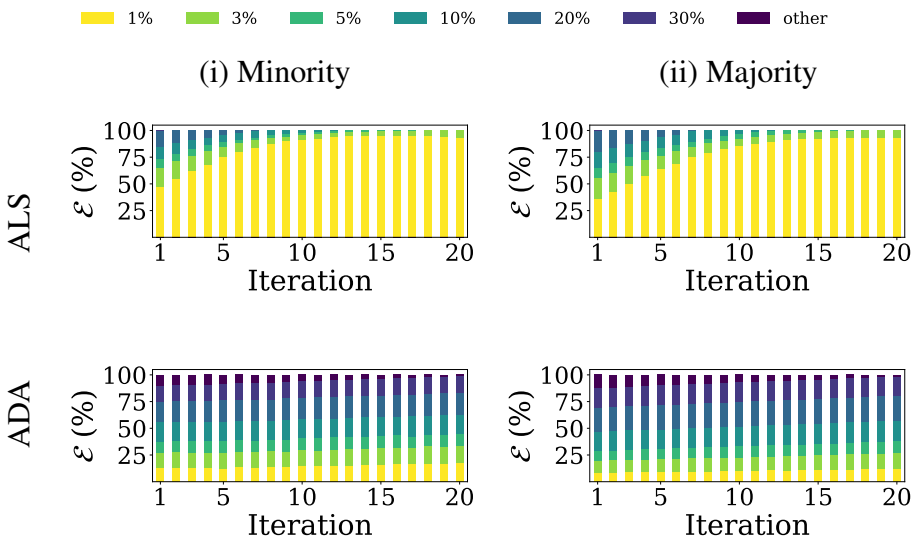
Figure 5.7: Gini coefficient computed on the in-degree of both minority and majority, for all the recommenders and networks, with B-PSB.

In Figure 5.8 we focus only on two recommenders, but results are consistent also with the other two models. In particular, we have similar findings for ALS and SLS, while ADA results really close to RND. Figure 5.8 shows that, with graphs presenting either a homophilic or a heterophilic minority (G1, G3), only a subset of nodes receives most of the exposure produced. What is also evident is that after each iteration, the number of nodes getting most of the exposure become smaller, confirming some recent analytical results that point out how rankings can be biased towards few individuals getting most of the exposure [GGLM19]. Specifically, this effect, in the long-term, results faster for two specific groups: the homophilic minority in G1 and the non-homophilic majority in G3. In the first case, after only 5 iterations the nodes belonging to the top-1% acquires more than 75% of exposure. While in G3, despite the majority group not being biased, after 15 iterations, only the top-3% of nodes is recommended. Moreover, in both graphs, ADA does not present the same increase in disparity in the long-term, but still, only a small fraction of users (20%) receives consistently the 75% of the exposure in both groups.

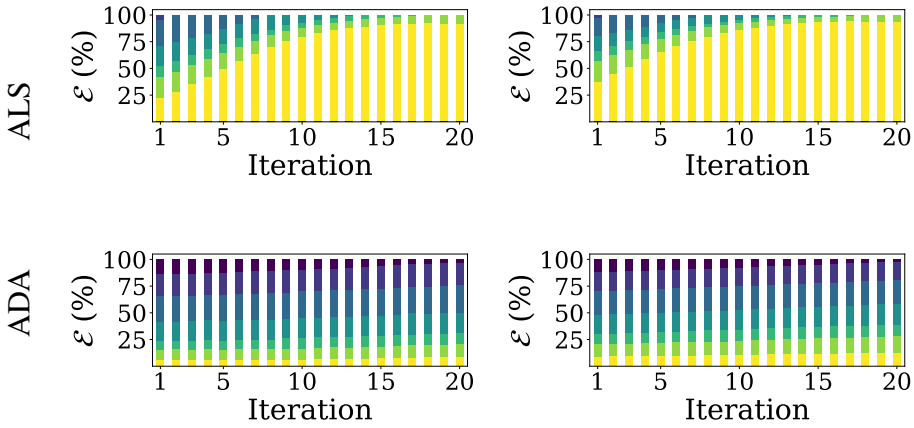
Observation 11. *When the minority presents non-neutral preferences (either homophilic or heterophilic), ALS and SLS can increase disparity in both exposure and in-degree: a small subset nodes benefits in terms of exposure by the injection of new links, and those are also the ones with highest degree. The cardinality of this subset of nodes becomes smaller after each iteration.*

5.4.4 Model Evaluation

In the experiments seen so far, we used the same configuration of k and α. To analyze how those input parameters may affect the simulation outcome, we next produce configurations presenting more sparse interactions (smaller α) and longer lists of recommendations (larger k). In Fig. 5.9 we present the exposure of the minority for G1, simulating the user behavior using B-PSB. In the first figure (Fig. 5.9a) we tune different values of α,



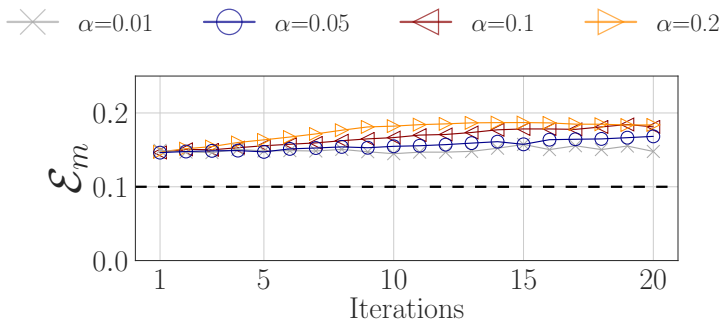
(a) G1



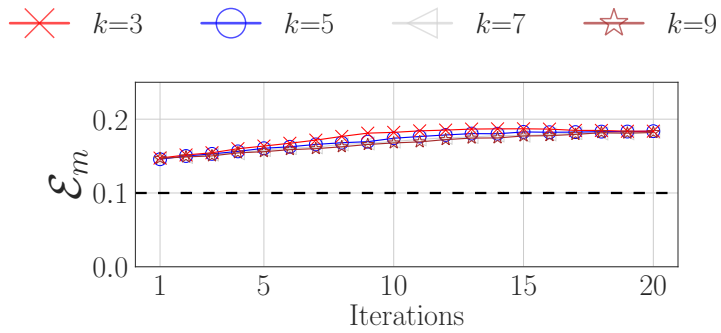
(b) G3

Figure 5.8: Distribution of Exposure among nodes, where each color delimitates the % of nodes with highest degree (best seen in color).

which the smaller, the less the number of users sampled to submit new recommendations. The plot shows how the exposure tends to growth, as expected, but with a slower pace. This parameter can be tuned looking at interactions generated by the social media platform over time. Here the length of recommendations is fixed to $k = 3$. In the case of longer lists of recommendations, the length of recommendation output impacts even less on the final output (Fig. 5.9b). The effect observed with the smaller recommendation list ($k = 3$) presents a trend close to all the other configurations. In this case, α is fixed to the usual value of 0.2.



(a) Level of Sparsity



(b) Length of Top-*k*

Figure 5.9: Testing different values of α and k on G1, with B-PSB.

5.5 Summary

The main goal of this work is to improve our comprehension of the long-term consequences on the disparate exposure of a minority in the recommendations provided by people recommender systems in a social network. In this endeavour we need to take care of the interplay between the recommender algorithm, user behaviour in accepting the recommendations, and pre-existing conditions in the network (e.g., the existence of an homophilic minority). Our analysis shows how the initial level of homophily within a subpopulation in the graph can drive exposure inequalities that grow over time: this is obtained without considering organic growth of the network (i.e., new links are created only if recommended by the algorithm) and assuming a homogenous user behavior for accepting or rejecting the link recommendations. Our work analyzes the impact of human biases, such as homophilic behavior, and link recommender algorithms on the disparate exposure of a minority at the level of the whole network.

Part III

Mitigating Bias in Recommender Systems

6

Rewiring What-to-Watch-Next Recommendations to Reduce Radicalization Pathways

6.1 Introduction

"*What-to-watch-next*" (W2W) recommenders are key features of video sharing platforms [ZHW+19], as they sustain user engagement, thus increasing content views and driving advertisement and monetization. However, recent studies have raised serious concerns about the potential role played by W2W recommenders, specifically in driving users towards undesired or polarizing content [LZ20]. Specifically, radicalized communities[1] on social networks and content sharing platforms have been recognized as keys in the consumption of news and in building opinions around politics and related subjects [Lew18, Roo19, WW18]. Recent work highlights the role of recommender systems, which may steer users towards radicalized content, eventually building "*radicalization pathways*" [Lew18,

[1] From [MM08]: "*Functionally, political radicalization is increased preparation for and commitment to intergroup conflict. Descriptively, radicalization means change in beliefs, feelings, and behaviors in directions that increasingly justify intergroup violence and demand sacrifice in defense of the ingroup.*"

ROW$^+$20] (i.e., a user might be further driven towards radicalized content even when this was not her initial intent). In this chapter, we study the problem of reducing the prevalence of radicalization pathways in W2W recommenders while maintaining the relevance of recommendations.

Formally, we model a W2W recommender system as a directed labeled graph where nodes correspond to videos (or other types of content) and directed edges represent recommendation links from one node to another[2]. In this scenario, each video is accompanied by the same number d of recommendation links, and thus every node in the graph has the same out-degree d. Moreover, each node has a binary label such as "harmful" (e.g., radicalized) or "neutral" (e.g., non-radicalized). The browsing activity of a user through the W2W recommendations is modeled as a *random walk* on the graph: after visiting a node (e.g., watching a video), the user moves to one of the d recommended videos with a probability that depends on its visibility or ranking in the recommendation list. In this setting, for each harmful node v, we measure the expected number of consecutive harmful nodes visited in a random walk before reaching any neutral node. We call this measure the "*segregation*" score of node v: intuitively, it quantifies how easy it is to get "stuck" in radicalization pathways starting from a given node. Our goal is to reduce the segregation of the graph while guaranteeing that the quality of recommendations is maintained, where the quality is measured by the *normalized discount cumulative gain* [BGW18, JK02] (nDCG) of each node. An important challenge is that the underlying recommendation graph has intrinsically some level of homophily because, given that the W2W seeks to recommend relevant videos, it is likely to link harmful nodes to other harmful nodes.

We formulate the problem of reducing the segregation of the graph as selecting k rewiring operations on edges (corresponding to modifications in the lists of recommended videos for some nodes) so as to minimize the maximum of segregation scores among all harmful nodes, while maintain-

[2]For ease of presentation, we focus on video sharing platforms. We note that the same type of recommendations occurs in many other contexts such as, for instance, news feeding platforms as shown in our experiments (see Section 7.5).

ing recommendation quality measured by nDCG above a given threshold for all nodes. We prove that our k-REWIRING problem is NP-hard and NP-hard to approximate within any factor. We therefore turn our attention to design efficient and effective heuristics. Our proposed algorithm is based on the *absorbing random walk theory* [MMG15], thanks to which we can efficiently compute the segregation score of each node and update it after every rewiring operation. Specifically, our method finds a set of k rewiring operations by greedily choosing the optimal rewiring for the special case of $k = 1$ – i.e., the 1-REWIRING problem, then updates the segregation score of each node. We further design a sorting and pruning strategy to avoid unnecessary attempts and thus improve the efficiency for searching the optimal rewiring. Though the worst-case time complexity of our algorithm is quadratic with respect to the number of nodes n, it exhibits much better performance (nearly linear w.r.t. n) in practice.

Finally, we present experiments on two real-world datasets: one in the context of video sharing and the other in the context of news feeds. We compare our proposed algorithm against several baselines, including an algorithm for suggesting new edges to reduce radicalization in Web graphs. The results show that our algorithm outperforms existing solutions in mitigating radicalization pathways in recommendation graphs.

In the rest of this chapter, we first review the literature relevant to our work in Section 6.2. Then, we introduce the background and formally define our problem in Section 6.3. Our proposed algorithms are presented in Section 6.4. The experimental setup and results are shown in Section 6.5. Finally, we conclude this chapter and discuss possible future directions in Section 6.6.

6.2 Related Work

A great deal of research has been recently published about the potential created by unprecedented opportunities to access information on the Web and social media. These risks include the spread of misinformation [AG17, SSW+17], the presence of bots [FVD+16], the abundance of

offensive hate speech [MZ17, MSB17], the availability of inappropriate videos targeting children [PPZ+20], the increase in controversy [GMGM16] and polarization [GJCK13], and the creation of radicalization pathways [ROW+20]. Consequently, a substantial research effort has been devoted to model, detect, quantify, reduce, and/or block such negative phenomena. We discuss here the existing studies that are the most relevant to our work here – in particular, algorithmic approaches to optimizing graph structures for achieving the aforementioned goals.

A line of research deals with limiting the spread of undesirable content in a social network via edge manipulation [KSM08, TPE+12, KTS+13, KDS14, SAPV15, LET15, YLW+19]. In these studies, the graph being manipulated is a network of users where the edges represent connections such as friendship or interactions among users. In contrast, we consider a graph of content items (e.g., videos or news), where the edges represent recommendation links. Moreover, these algorithmic methods are primarily based on information propagation models, while our work is based on random walks.

Another line of work aims at reducing controversy, disagreement, and polarization by edge manipulation in a social network, exposing users to others with different views [GMGM17, CLB18, MMT18, CM20b, HMRU21, IMR21]. A seminal work [GMGM17] introduces the *controversy score* of a graph based on random walks and propose an efficient algorithm to minimize it by edge addition. Similarly, a work by [MMT18] introduce the *Polarization-Disagreement index* of a graph based on Friedkin-Johnsen dynamics and propose a network-design approach to find a set of "best" edges that minimize this index. Another contribution defines the *worst-case conflict risk* and *average-case conflict risk* of a graph, also based on Friedkin-Johnsen dynamics, and propose algorithms to locally edit the graphs for reducing both measures [CLB18]. In the same direction, the work by [CM20b] analyzes the impact of "filter bubbles" in social network polarization and how to mitigate them by graph modification. Also, [IMR21] define a polarization reduction problem by adding edges between users from different groups and propose integer programming-based methods to solve it. Another related line of work proposes to

model and mitigate the disparate exposure generated by people recommenders (e.g. who-to-follow link predictions) in presence of effects like homophily and polarization [FCBC21, FBBC20, CMMB21, PKS20]. These studies also deal with networks of users, while in our case we consider a network of items.

The work probably most related to ours is the one by [HMRU21], which considers a graph of items (e.g., Web pages with hyperlinks) and defines the structural bias of a node as the difficulty/effort needed to reach nodes of a different opinion. They, then propose an efficient approximation algorithm to reduce the structural bias by edge insertions. There are three main differences between this and our work. First, two-directional edge manipulations (from class A to B and also from B to A) are considered by them [HMRU21], but one-directional edge manipulations (from harmful to neutral nodes only) are considered in our work. Second, they consider inserting new links on a node, which better fits the case of Web pages, but we consider rewiring existing edges, which better fits the case of W2W recommenders. Third, they define the structural bias of the graph as the sum of the bubble radii of all nodes, while we define the segregation of the graph as the worst-case segregation score among all harmful nodes. We compare our proposed algorithm with theirs in our experiments.

A recent line of work introduces the notion of *reachability* in recommender systems [DRR20, CDR21]. Instead of rewiring the links, they focus on making allowable modifications in the user's rating history to avoid unintended consequences such as filter bubbles and radicalization. However, as the problem formulation is different from ours, their proposed methods are not applicable to our problem.

Finally, there are many studies on modifying various graph characteristics, such as shortest paths [PBG11, PPT15], centrality [PPT16, CDSV16, MSS$^+$18, BCD$^+$18, DOS19, WWRM20], opinion dynamics [AS19, CFG20], and so on [CAT14, Pap15, LY15, ZCWL18], by edge manipulation.

6.3 Preliminaries

Let us consider a set V of n items and a matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, where each entry $s_{uv} \in [0, 1]$ at position (u, v) denotes the relevance score of an item v given that a user has browsed an item u. This expresses the likelihood that a user who has just watched u would be interested in watching v. Typically, a recommender system selects the d most relevant items to compose the recommendation list $\Gamma^+(u)$ of u, where the number of recommendations d is a design constraint (e.g., given by the size of the app window). We assume that the system selects the top-d items v w.r.t. s_{uv} and that their relevance score uniquely determines the ranking of the d items in $\Gamma^+(u)$. For each $v \in \Gamma^+(u)$, we use $i_u(v)$ to denote its ranking in $\Gamma^+(u)$. After a user has seen u, she/he will find the next item to see from $\Gamma^+(u)$, and the probability p_{uv} of selecting $v \in \Gamma^+(u)$ depends on the ranking $i_u(v)$ of v in $\Gamma^+(u)$. More formally, $p_{uv} = f(i_u(v))$, where f is a non-increasing function that maps from $i_u(v)$ to p_{uv} with $\sum_{v \in \Gamma^+(u)} p_{uv} = 1$.

This setting can be modeled as a directed probabilistic d-regular graph $G = (V, E, \mathbf{M})$, where the node set V corresponds to the set of all n items, the edge set E comprises $n \cdot d$ edges where each node $u \in V$ has d out-edges connected to the nodes in $\Gamma^+(u)$, and $\mathbf{M}$ is an $n \times n$ transition matrix with a value of p_{uv} for each $(u, v) \in E$ and 0 otherwise. A user's browsing session is thus modeled as a random walk on G starting from an arbitrary node in V with transition probability p_{uv} for each $(u, v) \in E$.

We further consider that the items in V are divided into two disjoint subsets V_n and V_h (i.e., $V_n \cap V_h = \emptyset$ and $V_n \cup V_h = V$) corresponding to "neutral" (e.g., not-radicalized) and "harmful" (e.g., radicalized) nodes, respectively.

The risk we want to mitigate is having users stuck in a long sequence of harmful nodes while performing a random walk. In order to quantify this phenomenon we define the measure of *segregation*. Given a set $S \subset V$ of nodes and a node $u \in V \setminus S$, we use a random variable $T_u(S)$ to indicate the first instant when a random walk starting from u reaches (or "hits") any node in S. We define $\mathbb{E}_G[T_u(S)]$ as the *hitting length* of u w.r.t. S, where the expectation is over the space of all possible random

walks on G starting from u. In our case, we define the segregation score z_u of node $u \in V_h$ by its expected hitting length $\mathbb{E}_G[T_u(V_n)]$ w.r.t. V_n. The segregation $Z(G)$ of graph G is defined by the maximum of segregation scores among all nodes in V_h – i.e., $Z(G) = \max_{u \in V_h} z_u$. In the following, we omit the argument G from $Z(G)$ when it is clear from the context.

Our main problem in this chapter is to mitigate the effect of segregation by modifying the structure of G. Specifically, we aim to find a set O of rewiring operations on G, each of which removes an existing edge $(u, v) \in E$ and inserts a new one $(u, w) \notin E$ instead, such that $Z(G^O)$ is minimized, where G^O is the new graph after performing O on G. For simplicity, we require that $u, v \in V_h$, $w \in V_n$, and $p_{uv} = p_{uw}$. In other words, each rewiring operation changes the recommendation list $\Gamma^+(u)$ of u by replacing one (harmful) item $v \in \Gamma^+(u)$ with another (neutral) item $w \notin \Gamma^+(u)$ and keeping the ranking $i_u(w)$ of w the same as the ranking $i_u(v)$ of v in $\Gamma^+(u)$.

Another goal, which is often conflicting, is to preserve the relevance of recommendations after performing the rewiring operations. Besides requiring only a predefined number k of rewirings, we also consider an additional constraint on the loss in the quality of the recommendations. For this purpose we adopt the well-known *normalized discounted cumulative gain* (nDCG) [JK02, BGW18] to evaluate the loss in the quality. Formally, the *discounted cumulative gain* (DCG) of a recommendation list $\Gamma^+(u)$ is defined as:

$$\texttt{DCG}(\Gamma^+(u)) = \sum_{v \in \Gamma^+(u)} \frac{s_{uv}}{1 + \log_2(1 + i_u(v))}$$

Then, we define the quality loss of $\Gamma^+(u)$ after rewiring operations by nDCG as follows:

$$L(\Gamma^+(u)) = \texttt{nDCG}(\Gamma^+(u)) = \frac{\texttt{DCG}(\Gamma^+(u))}{\texttt{DCG}(\Gamma_0^+(u))} \tag{6.1}$$

where $\Gamma_0^+(u)$ is the original (ideal) recommendation list where all the top-d items that are the most relevant to u are included.

Let $o = (u, v, w)$ be a rewiring operation that deletes (u, v) while adding (u, w) and O be a set of rewiring operations. For ease of presentation, we define a function $\Delta(O) \triangleq Z(G) - Z(G^O)$ to denote the decrease in the segregation after performing the rewiring operations in O and updating G to G^O. We are now ready to formally define the main problem studied in this chapter.

Problem 1 (k-REWIRING)**.** *Given a directed probabilistic graph $G = (V, E, \mathbf{M})$, a positive integer $k \in \mathbb{Z}^+$, and a threshold $\tau \in (0, 1)$, find a set O of k rewiring operations that maximizes $\Delta(O)$, under the constraint that $L(\Gamma^+(u)) \geq \tau$ for each node $u \in V$.*

The hardness of the k-REWIRING problem is analyzed in the following theorem.

Theorem 1. *The k-REWIRING problem is NP-hard and NP-hard to approximate within any factor.*

We show the NP-hardness of the k-REWIRING problem by reducing from the VERTEXCOVER problem. Furthermore, we show that finding an α-approximate solution of the k-REWIRING problem for any factor $\alpha > 0$ is at least as hard as finding the minimum vertex cover of a graph. Therefore, the k-REWIRING problem is NP-hard to approximate within any factor.

6.3.1 Proof of Theorem 1

Proof. We prove the NP-hardness of the k-REWIRING problem by a reduction from the VERTEXCOVER problem [GJ79].

A VERTEXCOVER instance is specified by an undirected graph $G = (V, E)$, where $|V| = n$ and $|E| = m$, and an integer k. It asks whether G has a vertex cover of size at most k, i.e., whether there exists a subset $C \subseteq V$ with $|C| \leq k$ such that $\{v_i, v_j\} \cap C \neq \varnothing$ for every edge $e = (v_i, v_j) \in E$. We construct an instance of the k-REWIRING problem on G^* from a VERTEXCOVER instance on G as illustrated in Figure 6.1a. Given a graph $G = (V, E)$, the graph $G^* = (V^*, E^*)$ is constructed as follows:

One vertex in G^* is created for each $e \in E$ and $v \in V$. Furthermore, four vertices h_1, h_2, n_1, n_2 are added to G^*. Let $V_h^* = E \cup V \cup \{h_1, h_2\}$ be the set of $(m + n + 2)$ "harmful" vertices (in red) and $V_n^* = \{n_1, n_2\}$ be the set of two "neutral" vertices (in blue). Then, for each $e = (v_i, v_j) \in E$, two directed edges (e, v_i) and (e, v_j) are added to G^*. For each $v \in V$, two directed edges (v, h_1) and (v, h_2) are added to G^*. Finally, four directed edges (h_1, n_1), (h_1, n_2), (h_2, n_1), and (h_2, n_2) are added to G^*. The out-degree d of each red node in G^* is 2. Accordingly, the transition probability of every edge in G^* is set to 0.5.

We first show that there will be a set O of at most k rewiring operations such that $\Delta(O) > 0$ after the rewiring operations in O are performed on G^* *if* G has a vertex cover of size at most k. For the original G^*, we have $z(h_1) = z(h_2) = 1$, $z(v) = 2$ for each vertex $v \in V$, and $z(e) = 3$ for each edge $e \in E$. Thus, we have $Z = z(e) = 3$. So, we will have $\Delta(O) > 0$ as long as $z'(e) < 3$ for each edge $e \in E$. Let $C = \{v_1, \ldots, v_k\}$ be a size-k vertex cover of G. We construct a set $O = \{o_1, \ldots, o_k\}$ of k rewiring operations on G^*, where $o_i = (v_i, h_1, n_1)$, corresponding to C, as illustrated in Figure 6.1b. After performing the set O of rewiring operations on G^*, we have two cases for $z'(e)$ of each $e = (v_i, v_j)$:

$$z'(e) = \begin{cases} 0.5 \times 3 + 0.5 \times 2 = 2.5, & \textit{if } |\{v_i, v_j\} \cap C| = 2 \\ 0.5 \times 3 + 0.5 \times 2.5 = 2.75, & \textit{if } |\{v_i, v_j\} \cap C| = 1 \end{cases}$$

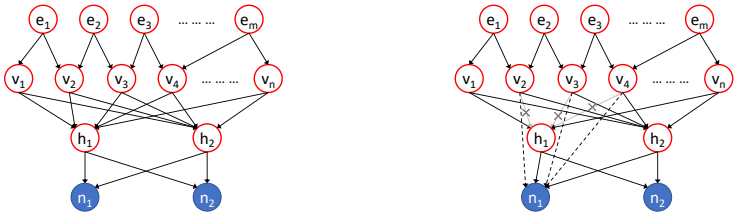
Since C is a vertex cover, there is no edge $e = (v_i, v_j)$ such that $\{v_i, v_j\} \cap C = \varnothing$. Therefore, after performing the set O of rewiring operations on G^*, it must hold that $z'(e) < 3$ for every $e \in E$ and thus $\Delta(O) > 0$.

We then show that there will be a set O of at most k rewiring operations such that $\Delta(O) > 0$ after the rewiring operations in O are performed on G^* *only if* G has a vertex cover of size at most k. Or equivalently, if G does not have a vertex cover of size k, then any set O of k rewiring operations performed on G^* cannot make $\Delta(O) > 0$. Since G does not have a vertex cover of size k, there must exist some edge $\overline{e} = (v_i, v_j)$ with $\{v_i, v_j\} \cap \overline{C} = \varnothing$ for any size-k vertex set $\overline{C} \subseteq V$. Therefore, after performing the set $\overline{O}$ of k rewiring operations corresponding to $\overline{C}$, we

have $z'(\overline{e}) = 3$ for an uncovered edge $\overline{e}$. So, we can say that any set of k rewiring operations from V cannot make $\Delta(O) > 0$. Furthermore, we consider the case of k rewiring operations from E, i.e., to find a set of k edges $\{e_1, \ldots, e_k\}$ and rewire one out-edge from each of them to n_1 or n_2. In this case, we can always find some unselected edge $\overline{e}$ with $z'(\overline{e}) = 3$ as long as $m > k$, which obviously holds as G does not have a vertex cover of size k. Finally, we consider the case of a "hybrid" set of k rewiring operations from both E and V. W.l.o.g., we assume that there are $(k - k')$ operations from V and k' operations from E for some $0 < k' < k$. Since G does not have a vertex cover of size k, we can say that any vertex set $\overline{C}$ of size $(k-k')$ can cover at most $(m-k'-1)$ edges. Otherwise, we would find a vertex cover of size k by adding k' vertices to cover the remaining k' edges and thus lead to contradiction. Therefore, after performing only k' rewiring operations from E, there always exists at least one edge $\overline{e}$ that are covered by neither the vertex set nor the edge set, and thus $z'(\overline{e}) = 3$ and $\Delta(O) = 0$. Considering all the three cases, we prove that any set of k rewiring operations performed on G^* cannot make $\Delta(O) > 0$ if G does not have a vertex cover of size k.

Given that both the "*if*" and "*only-if*" directions are proven and G^* can be constructed from G in $O(m+n)$ time, we reduce from the VERTEX-COVER problem to the k-REWIRING problem in polynomial time and thus prove that the k-REWIRING problem is NP-hard.

To show the hardness of approximation, we suppose that there is a polynomial-time algorithm $\mathcal{A}$ that approximates the k-REWIRING problem within a factor of $\alpha > 0$. Or equivalently, for any k-REWIRING instance, if O^* is the set of k optimal rewiring operations, then the set O' of k rewiring operations returned by $\mathcal{A}$ will always satisfy that $\Delta(O') \geq \alpha \cdot \Delta(O^*)$. Let us consider a k-REWIRING instance on the above graph G^* constructed from G and k be the size of the minimum vertex cover of G. For this instance, the optimal solution O^* of the k-REWIRING problem exactly corresponds to the minimum vertex cover C^* of G with $\Delta(O^*) > 0$; any other solution O' will lead to $\Delta(O') = 0$, as we have shown in this proof. If $\mathcal{A}$ could find a solution for the k-REWIRING problem with any approximation factor $\alpha > 0$ in polynomial time, then $\mathcal{A}$ would have



(a) Construct G^* from G (b) k-REWIRING on G^*

Figure 6.1: Illustration of the reduction from the VERTEXCOVER problem to the k-REWIRING problem.

solved the VERTEXCOVER problem in polynomial time, which has been known to be impossible unless P=NP. Therefore, the k-REWIRING problem is NP-hard to approximate with any factor. □

6.3.2 Absorbing Random Walk

We now provide notions from the absorbing random walk theory [MMG15] on which our algorithms are built.

The k-REWIRING problem asks to minimize *segregation*, which is defined as the maximum hitting length from any harmful node to neutral nodes. Specifically, in the context of k-REWIRING for the given probabilistic directed graph $G = (V, E, \mathbf{M})$, we equivalently consider a modified transition matrix $\mathbf{M}$ as follows:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{hh} & \mathbf{M}_{hn} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

In the matrix $\mathbf{M}$ above, each *neutral* node has been set to be *absorbing*, i.e., its transition probability to itself is set to $p_{ii} = 1$ and 0 to other nodes (see the bottom row of $\mathbf{M}$). Intuitively, no random walk passing through an absorbing node can move away from it [MMG15]. For each *harmful* node, its transition probabilities remain unmodified (see the top row of $\mathbf{M}$) and thus the node remains *transient* (i.e., *non-absorbing*).

The fundamental matrix $\mathbf{F}$ can be computed from the sub-matrix $\mathbf{M}_{hh}$ as follows [MMG15]:

$$\mathbf{F} = (\mathbf{I} - \mathbf{M}_{hh})^{-1}$$

where the entry f_{uv} represents the expected total number of times that the random walk visits node v having started from node u. Then, the expected length of a random walk that starts from any node and stops when it gets absorbed is given by vector $\mathbf{z}$:

$$\mathbf{z} = \begin{bmatrix} (\mathbf{I} - \mathbf{M}_{hh})^{-1} \\ \mathbf{0} \end{bmatrix} \mathbf{1} \tag{6.2}$$

where $\mathbf{1}$ is an n-dimensional vector of all 1's. Here, the i-th entry z_i of vector $\mathbf{z}$ represents the expected number of random walk steps before being absorbed by any absorbing node, assuming that the random walk starts from the i-th node.

Given that the absorbing and transient nodes are set to correspond exactly to the neutral and harmful nodes, respectively, the values of $\mathbf{z}$ correspond exactly to the expected hitting length as used to define segregation. Hence, the k-REWIRING problem asks to choose a set of k rewiring operations to minimize the maximum entry $Z = \max_{1 \leq i \leq n} z_i$ of vector $\mathbf{z}$.

6.4 Algorithms

Since k-REWIRING is NP-hard to approximate within any factor, we propose an efficient heuristic. The heuristic is motivated by the following observation: despite the NP-hardness of k-REWIRING, its special case when $k = 1$, which we call 1-REWIRING, is solvable in polynomial time. Given an optimal 1-REWIRING algorithm, k-REWIRING can be addressed by running it k times.

We begin our presentation of algorithms by showing a brute-force algorithm for finding the optimal solution of 1-REWIRING (Section 6.4.1), as well as a way to speed it up via incremental updates (Section 6.4.2). Subsequently, we propose our optimal 1-REWIRING algorithm that improves the efficiency of the brute-force algorithm by faster rewiring search

(Section 6.4.3). Finally, we present how our 1-REWIRING algorithm is used for k-REWIRING (Section 6.4.4).

6.4.1 Brute-Force Algorithm for 1-REWIRING

Given a graph G and a rewiring operation o, we use $\Delta(o)$ to denote the decrease in Z after performing o on G. We present a brute-force algorithm to find the rewiring operation o^* that maximizes $\Delta(o)$. The algorithm has three steps: (1) enumerate the set Ω of all feasible rewiring operations for G and a given threshold τ; (2) get $\Delta(o)$ for each $o \in \Omega$ by computing Z using Eq. 6.2 on G before/after performing o; (3) find the operation o that has the largest $\Delta(o)$ as the optimal solution o^*. In the brute-force algorithm, since the number of existing edges is $O(dn)$ and the number of possible new edges to rewire is $O(n)$ for each existing edge, the size of Ω is $O(dn^2)$. In addition, the old and new values of Z can be computed by matrix inversion using Eq. 6.2 in $O(n^3)$ time. Therefore, the brute-force algorithm runs in $O(dn^5)$ time. As all feasible operations are examined, this solution is guaranteed to be optimal.

The brute-force algorithm is impractical if the graph is large, due to the huge number of feasible operations and the high cost of computing Z. We introduce two strategies to improve its efficiency. First, we update the vector $\mathbf{z}$ incrementally for a rewiring operation. Second, we devise efficient strategies to avoid unnecessary computation when searching for the optimal rewiring operation, leading to our optimal 1-REWIRING algorithm.

6.4.2 Incremental Update of Vector z

We analyze how the fundamental matrix $\mathbf{F}$ and vector $\mathbf{z}$ change after performing a rewiring operation $o = (u, v, w)$. Two edits will be performed on G for o: (1) the removal of an existing edge $(u, v) \in E$ and (2) the insertion of a new edge $(u, w) \notin E$ to E.

The two operations update the transition matrix $\mathbf{M}$ to $\mathbf{M}'$ as follows:

$$\mathbf{M}' = \mathbf{M} + \mathbf{e}\mathbf{g}^\top$$

where $\mathbf{e}$ is an n-dimensional column vector that indicates the position of the source node u:

$$e_j = \begin{cases} 1 & \text{if } j = u \\ 0 & \text{otherwise} \end{cases}$$

and $\mathbf{g}^\top$ is an n-dimensional row vector that denotes the changes in the transition probabilities. Specifically, for the removal of (u, v) and insertion of (u, w), the probability p_{uv} of (u, v) is reassigned to (u, w). We denote the probability as $p_o = p_{uv}$. Formally,

$$g_j = \begin{cases} -p_o & \text{if } j = v \\ +p_o & \text{if } j = w \\ 0 & \text{otherwise} \end{cases}$$

Thus, operation $o = (u, v, w)$ on the fundamental matrix $\mathbf{F}$ yields an updated fundamental matrix $\mathbf{F}'$:

$$\mathbf{F}' = ((\mathbf{I} - \mathbf{M}_{hh}) - \mathbf{e}\mathbf{g}^\top)^{-1} = (\mathbf{F}^{-1} + (-1)\mathbf{e}\mathbf{g}^\top)^{-1}$$

By applying the Sherman-Morrison formula [PTVF07], we can avoid the computation of the new inverse and express $\mathbf{F}'$ as:

$$\mathbf{F}' = \mathbf{F} - \frac{\mathbf{F}\mathbf{e}\mathbf{g}^\top\mathbf{F}}{1 + \mathbf{g}^\top\mathbf{F}\mathbf{e}} \tag{6.3}$$

Accordingly, the new vector $\mathbf{z}'$ is expressed as:

$$\mathbf{z}' = \mathbf{z} - \frac{\mathbf{F}\mathbf{e}\mathbf{g}^\top\mathbf{F}}{1 + \mathbf{g}^\top\mathbf{F}\mathbf{e}}\mathbf{1} \tag{6.4}$$

The denominator of the second term in Eq. 6.4 can be written as:

$$1 + \mathbf{g}^\top\mathbf{F}\mathbf{e} = 1 - p_o(f_{wu} \cdot \mathbf{1}_{w \in V_h} - f_{vu})$$

where $\mathbf{1}_{w \in V_h}$ is an indicator that is equal to 1 if $w \in V_h$ and 0 otherwise. Because, as mentioned in Section 6.3, we restrict ourselves to rewiring with $w \notin V_h$, the above expression is simplified as:

$$1 + \mathbf{g}^\top\mathbf{F}\mathbf{e} = 1 + p_o f_{vu}.$$

Meanwhile, the numerator of the second term in Eq. 6.4 is written as:

$$\mathbf{F}\mathbf{e}\mathbf{g}^\top\mathbf{F}\mathbf{1} = -\mathbf{f}_u(z_w \cdot \mathbf{1}_{w\in V_h} - z_v)p_o$$

where $\mathbf{f}_u$ is the column vector corresponding to u in $\mathbf{F}$, z_w and z_v are the entries of $\mathbf{z}$ for u and v, respectively. As previously, because $w \notin V_h$, we have that Eq. 6.4 is simplified as:

$$\mathbf{z}' = \mathbf{z} - \frac{\mathbf{f}_u z_v}{1/p_o + f_{vu}}.$$

For any harmful node h, we calculate its decrease $\Delta(h, o)$ in segregation score after performing $o = (u, v, w)$ as:

$$\Delta(h, o) = z_h - z'_h = \frac{f_{hu} z_v}{1/p_o + f_{vu}} \tag{6.5}$$

The optimal 1-REWIRING we present next is based on Eq. 6.5.

6.4.3 Optimal 1-REWIRING Algorithm

We now introduce our method to find the optimal solution o^* of 1-REWIRING, i.e., the rewiring operation that maximizes $\Delta(o)$ among all $o \in \Omega$. The detailed procedure is presented in Algorithm 1, to which the fundamental matrix $\mathbf{F}$ and segregation vector $\mathbf{z}$ are given as input. The algorithm proceeds in two steps: (1) candidate generation, as described in Lines 2–5, which returns a set Ω of possible rewiring operations that definitely include the optimal 1-REWIRING, and (2) optimal rewiring search, as described in Lines 6–16, which computes the objective value for each candidate rewiring to identify the optimal one. Compared with the brute-force algorithm, this method reduces the cost of computing $\Delta(o)$ since it only probes a few nodes with the largest segregation scores. In addition, it can still be guaranteed to find the optimal solution, as all rewiring operations that might be the optimal one have been considered.

Candidate generation. The purpose of this step is to exclude from enumeration all rewiring operations that violate the quality constraint. Towards this end, we do not consider any rewiring operation that for any

Algorithm 1: OPTIMAL 1-REWIRING

Input : Graph $G = (V, E, \mathbf{M})$, fundamental matrix $\mathbf{F}$, segregation vector $\mathbf{z}$, threshold τ

Output : Optimal rewiring operation o^*

Initialize $\Omega \leftarrow \emptyset$, $o^* \leftarrow NULL$, $\Delta^* \leftarrow 0$;
foreach *node* $u \in V_h$ **do**
    Find node $w \in V_n$ s.t. $(u, w) \notin E$ and s_{uw} is the maximum;
    foreach *node* $v \in V_h$ *with* $(u, v) \in E$ **do**
        Add $o = (u, v, w)$ to Ω if $L(\Gamma^+(u)) \geq \tau$ after replacing (u, v) with (u, w);
Sort nodes in V_h as $\langle h_1, \ldots, h_{n_h} \rangle$ in descending order of z_h;
foreach $o \in \Omega$ **do**
    Compute $\Delta(h_1, o)$ using Eq. 6.5;
    if $z'_{h_1} > z_{h_2}$ **then**
        $\Delta(o) \leftarrow \Delta(h_1, o)$;
    else
        Find the largest $j > 1$ such that $z'_{h_1} < z_{h_j}$;
        Compute $\Delta(h_i, o)$ for each $i = 2, \ldots, j$;
        $\Delta(o) \leftarrow z_{h_1} - \max_{i \in [1,j]} z'_{h_i}$;
    if $\Delta(o) > \Delta^*$ **then**
        $o^* \leftarrow o$ and $\Delta^* \leftarrow \Delta(o)$;
return o^*;

node u will lead to the *discount cumulative gain* (DCG) of u below the threshold τ. According to Eq. 6.5, we find that $\Delta(h, o)$ of node h w.r.t. $o = (u, v, w)$ is independent of (u, w). Therefore, for a specific node u, we can fix w to the neutral (absorbing) node with the highest relevance score s_{uw} and $(u, w) \notin E$ so that as many rewiring operations as possible are feasible. Then, we should select the node v where $(u, v) \in E$ will be replaced. We need to guarantee that $L(\Gamma^+(u)) \geq \tau$ after (u, v) is replaced by (u, w). For each node $v \in \Gamma^+(u)$, we can take s_{uv} and s_{uw} into Eq. 6.1. If $L(\Gamma^+(u)) \geq \tau$, we will list $o = (u, v, w)$ as a candidate. After consid-

ering each node $u \in V_h$, we generate the set Ω of all candidate rewiring operations.

Optimal rewiring search. The second step is to search for the optimal rewiring operation o^* from Ω. We first sort all harmful nodes in descending order of their segregation scores as $\langle h_1, h_2, \ldots, h_{n_h} \rangle$, where h_i is the node with the i-th largest segregation score. Since we are interested in minimizing the maximum segregation, we can focus on the first few nodes with the largest segregation scores and ignore the remaining ones. We need to compute $\Delta(o)$ for each $o \in \Omega$ and always keep the maximum of $\Delta(o)$. After evaluating every $o \in \Omega$, it is obvious that the one maximizing $\Delta(o)$ is o^*. Furthermore, to compute $\Delta(o)$ for some operation o, we perform the following steps: (1) compute $\Delta(h_1, o)$ using Eq. 6.5; (2) if $z'_{h_1} > z_{h_2}$, then $\Delta(o) = \Delta(h_1, o)$; (3) otherwise, find the largest j such that $z'_{h_1} < z_{h_j}$, compute $\Delta(h_i, o)$ for each $i = 2, \ldots, j$; in this case, we have $\Delta(o) = z_{h_1} - \max_{i \in [1,j]} z'_{h_i}$.

Time complexity. Compared with the brute-force algorithm, the size of Ω is reduced from $O(dn^2)$ to $O(dn)$. Then, sorting the nodes in V_h takes $O(n \log n)$ time. Moreover, it takes $O(1)$ time to compute $\Delta(h, o)$ for each h and o. For each $o \in \Omega$, $\Delta(h, o)$ is computed $O(n)$ times in the worst case. Therefore, the time complexity is $O(dn^2)$ in the worst case. However, in our experimental evaluation, we find that $\Delta(h, o)$ is computed only a small number of times. Therefore, if computing $\Delta(o)$ takes $O(1)$ time in practice, then the anticipated running time is $O\big(n(d + \log n)\big)$, as confirmed empirically.

6.4.4 Heuristic k-REWIRING Algorithm

Our k-REWIRING algorithm based on the 1-REWIRING algorithm is presented in Algorithm 2. Its basic idea is to find the k rewiring operations by running the 1-REWIRING algorithm k times. The first step is to initialize the fundamental matrix $\mathbf{F}$ and segregation vector $\mathbf{z}$. In our implementation, instead of performing the expensive matrix inversion (in Eq. 6.2), $\mathbf{F}$ and $\mathbf{z}$ are approximated through the power iteration method in [MMG15]. Then, the procedure of candidate generation is the same

Algorithm 2: HEURISTIC k-REWIRING

Input : Graph $G = (V, E, \mathbf{M})$, threshold τ, size constraint k
Output : A set O of k rewiring operations

```
1  Compute the initial F and z based on M;
2  Acquire Ω using Lines 2–5 of Algorithm 1;
3  Initialize O ← ∅;
4  for i ← 1, 2, ..., k do
5      Run Lines 6–16 of Algorithm 1 to get o* = (u*, v*, w*);
6      O ← O ∪ {o*};
7      Update G, M, F, and z for o*;
8      Delete the existing rewiring operations of u* from Ω and add new
         possible operations of u* to Ω;
9      if Ω = ∅ then
10         break;
11 return O;
```

as that in Algorithm 1. Next, it runs k iterations for getting k rewiring operations. At each iteration, it also searches for the the optimal rewiring operation $o^* = (u^*, v^*, w^*)$ among Ω as Algorithm 1. After that, G, $\mathbf{M}$, $\mathbf{F}$, and $\mathbf{z}$ are updated according to o^* (see Eq. 6.3 and 6.4 for the update of $\mathbf{F}$ and $\mathbf{z}$). Since the existing rewiring operations of u^* are not feasible anymore, it will regenerate new possible operations of u^* based on the updated $\Gamma^+(u^*)$ and the threshold τ to replace the old ones. Finally, the algorithm terminates when k rewiring operations have been found or there is no feasible operation anymore.

Time complexity. The time complexity of computing $\mathbf{F}$ and $\mathbf{z}$ is $O(\mathrm{iter} \cdot dn)$ where iter is the number of iterations in the power method. The time to update $\mathbf{F}$ and $\mathbf{z}$ for each rewiring operation is $O(n)$. Overall, its time complexity is $O(kdn^2)$ since it is safe to consider that $\mathrm{iter} \ll n$. In practice, it takes $O(1)$ time to compute $\Delta(o)$ and $\mathrm{iter} = O(k)$, and thus the running time of the k-REWIRING algorithm can be regarded as $O\left(kn\left(d + \log n\right)\right)$.

6.5 Experiments

Our experiments aim to: (1) show the effectiveness of our algorithm on mitigating radicalization pathways compared to existing algorithms; (2) test the robustness of our algorithm with respect to different thresholds τ; and (3) illustrate how much our algorithm can reduce the total exposure to harmful content.

6.5.1 Experimental Setup

Datasets. We perform experiments within two application domains: video sharing and news feeding.

For the first application, we use the **YouTube** dataset [ROW$^+$20], which contains 330,925 videos and 2,474,044 recommendations. The dataset includes node labels such as "*alt-right*", "*alt-lite*", "*intellectual dark web*" and "*neutral*". We categorize the first three classes as "radicalized" or harmful and the last class as "neutral," following the analysis done by this dataset's curators [ROW$^+$20], in which these three classes are shown to be overlapping in terms of audience and content. When generating the recommendation graphs, we consider only videos having a minimum of 10k views. In this way, we filter out all the ones with too few interactions. We consider the video-to-video recommendations collected via simulations as implicit feedback interactions, where the video-to-video interactions can be formatted as a square matrix, with position (u, v) containing the number of times the user jumped from video u to video v. Using alternating least squares (ALS) [HKV08], we can first derive the latent dimensions of the matrix, generate the scores (normalized to $[0, 1]$) and then build the recommendation lists for each video. We eventually create different d-regular graphs with $d \in \{5, 10, 20\}$. To evaluate the effect of graph size on performance, we also use a smaller subset of videos with only 100k or more views for graph construction. Finally, we have 3 smaller (**YT-D5-S**, **YT-D10-S**, and **YT-D20-S**) and 3 larger (**YT-D5-B**, **YT-D10-B**, and **YT-D20-B**) recommendation graphs.

For the second application, we use the **NELA-GT** dataset [NHA19],

which is a collection of 713k news in English. Each news article includes title, text, and timestamp, as well as credibility labels (reliable or unreliable). Our task is to reduce the risk of users getting stuck in unreliable content via *"what-to-read-next"* recommendations. To build the recommendation graphs, we compute the pairwise semantic similarities between news through the pre-generated weights with RoBERTa [LOG+19]. After normalizing the scores in the range $[0, 1]$, in order to reproduce different instances of news feeding websites, we generate different subsets of news by month. We perform our experiments on the 4 months with the largest number of news: August (**NEWS-1**), September (**NEWS-2**), October (**NEWS-3**) and November (**NEWS-4**).

Table 6.1: Characteristics of the recommendation graphs used in the experiments, including out-degree d, number of nodes n, number of edges m, fraction of nodes from V_h (i.e., n_h/n), and initial segregation Z^0 of each graph.

YouTube					
Name	d	n	m	n_h/n	Z^0
YT-D5-S	5	31524	157620	0.48	588.86
YT-D5-B		105143	525715	0.43	598.32
YT-D10-S	10	31524	315240	0.48	718.92
YT-D10-B		105143	1051430	0.43	718.37
YT-D20-S	20	31524	630480	0.48	328.03
YT-D20-B		105143	2102860	0.43	331.09
NELA-GT					
Name	d	n	m	n_h/n	Z^0
NEWS-1	10	27286	272860	0.61	88.53
NEWS-2		22296	222960	0.62	29.90
NEWS-3		28861	288610	0.61	335.23
NEWS-4		26114	261140	0.65	75.15

The characteristics of the ten recommendation graphs used in our experiments are reported in Table 6.1.

Algorithms. We compare our proposed heuristic (**HEU**) algorithm for k-REWIRING with three baselines and one existing algorithm. The first baseline (**BSL-1**) selects the set of k rewiring operations by running Algorithm 1. Instead of picking only one rewiring operation, it picks the k operations with the largest values of Δ all at once. The second baseline (**BSL-2**) considers the best possible k rewiring operations by looking at the initial values of the vector $\mathbf{z}$. It firsts select the k nodes with the largest z values, then among the possible rewiring operations from those nodes, it returns the k operations with the largest values of Δ. The third baseline (**RND**) just picks k random rewiring operations from all the candidates. Finally, the existing method we compare with is the *RePBubLik* algorithm [HMRU21] (**RBL**). It reduces the structural bias of the graph by looking at the bubble radius of the two partitions of nodes, returning a list of k new edges to add. The original algorithm is designed for the insertion of new links, and not for the rewiring (deletion + insertion). Consequently, we adapt the *RePBubLik* algorithm to our objective as follows: (1) we run it to return a list of potential edges to be added for reducing the structural bias of the harmful nodes; (2) for each potential insertion, in order to generate a rewiring operation, we check among the existing edges to find the one edge that meets the quality constraint τ after being replaced by the new edge; (3) we finally select a set of k rewiring operations from the previous step.

The experiments were conducted on a server running Ubuntu 16.04 with an Intel Broadwell 2.40GHz CPU and 29GB of memory. Our algorithm and baselines were implemented in Python 3[3].

6.5.2 Experimental Results

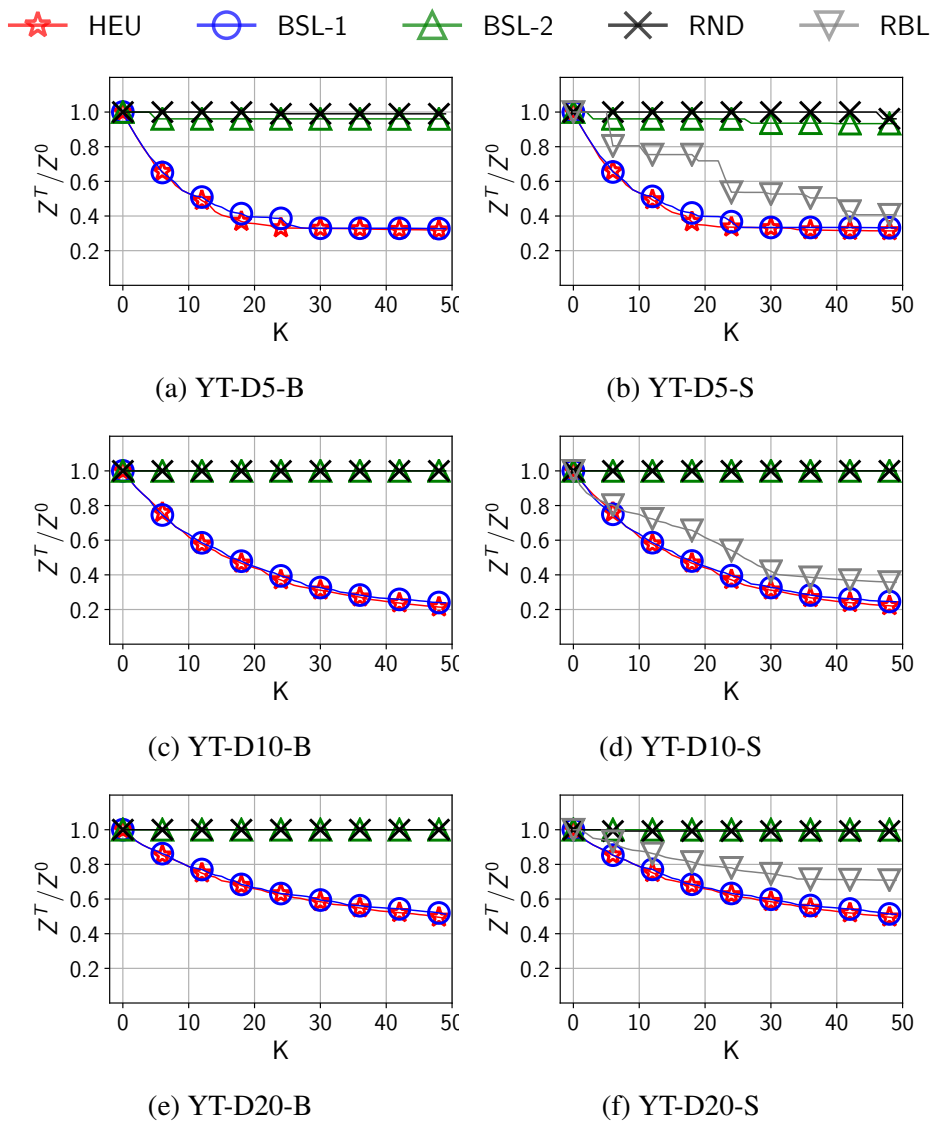
Effectiveness of our method. In Figure 6.2, we present the results on the YouTube recommendation graphs. On each graph, we evaluate the perfor-

[3]Our code and datasets are publicly available at https://github.com/FraFabbri/rewiring-what-to-watch

mance of each algorithm along 50 rewiring operations with the threshold of quality constraint is fixed to $\tau = 0.9$. We keep track of the relative decrease in the segregation Z^T/Z^0 after each rewiring operation, where Z^0 is the initial segregation and Z^T is the segregation after T rewiring operations. On all the graphs, it is clear that our heuristic algorithm (**HEU**) outperforms all the competitors. On the graphs with the smallest out-degree ($d = 5$), it decreases Z by over 40% within only 10 rewiring operations (i.e., $Z^{10}/Z^0 \leq 0.6$). In this case, it stops decreasing Z after 30 rewiring operations, which implies that only after a few rewiring operations our heuristic algorithm has found the best possible operations constrained by the threshold τ. On the graphs with $d = 10$, our heuristic algorithm is able to decrease Z by nearly 80%, which is even larger than the case of $d = 5$. This result is consistent in both smaller (YT-D10-S) and bigger (YT-D10-B) graphs. On the graphs with the largest out-degree ($d = 20$), the algorithm is still effective but, as expected, achieves a comparable reduction in Z after 50 operations.

The first baseline (**BSL-1**) shows almost the same solution quality as **HEU**, since most of the operations found by both algorithms are the same. Although the rewiring operations provided by *RePBubLik* (**RBL**) also decrease the original Z_0 significantly, they are less effective than the ones given by our algorithm. Also, with a smaller size of recommendation list ($d = 5$), it reaches some steady states along the iterations, where the new rewiring operations do not decrease the Z value at all. For the YouTube dataset, we present only the results of **RBL** on the smaller graphs (the second column of Figure 6.2), since it cannot finish in reasonable time (24 hours) on larger graphs. The other baseline (**BSL-2**) and the random solution (**RND**) do not produce substantial decreases over the initial Z_0.

In Figure 6.3, we present the results on the NELA-GT recommendation graphs. We also fix $\tau = 0.9$ in these experiments. Given that the values of Z^0 are smaller in the news recommendation graph, we evaluate the performance of different algorithms with smaller k (i.e., $k = 20$). As for the previous case, our heuristic algorithm is the one achieving the best performance on every graph, which reduces Z by at least 60% after 20 rewiring operations. Furthermore, on the graph with the biggest Z value



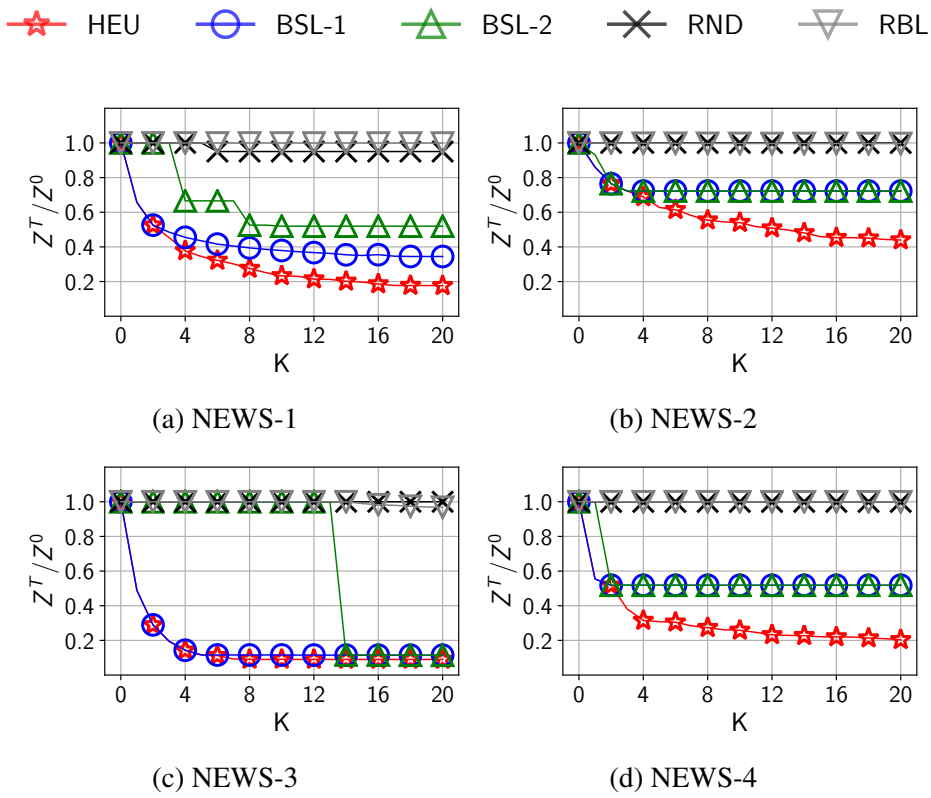
(a) YT-D5-B (b) YT-D5-S

(c) YT-D10-B (d) YT-D10-S

(e) YT-D20-B (f) YT-D20-S

Figure 6.2: Performance comparison in the YouTube dataset.

(NEWS-3), it decreases the initial segregation by more than 80% only after 4 rewiring operations. The two baselines (**BSL-1** and **BSL-2**) show



(a) NEWS-1

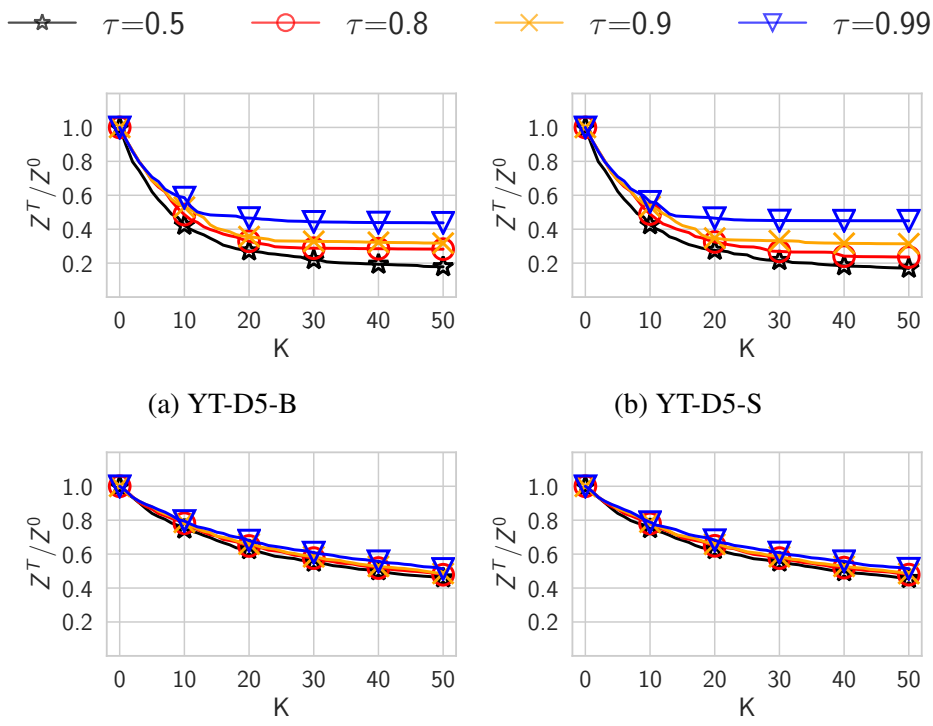
(b) NEWS-2

(c) NEWS-3

(d) NEWS-4

Figure 6.3: Performance comparison in the NELA-GT dataset.

comparable performance, but only on NEWS-3 they obtain close drops in Z_0 to **HEU** after 20 iterations. In the other cases, they are stuck in steady states far from **HEU**. The rewiring provided by *RePBubLik* (**RBL**) shows no significant decrease over the initial Z_0, which is comparable only to **RND**. The difference in performance between YouTube and NELA-GT can be to some extent attributed to differences in their degree distributions. We compute the Gini coefficient of the in-degree distribution of the graphs: for the YouTube graphs the Gini coefficient of in-degree for the harmful nodes is never below 90%; while for the NELA-GT graphs this index is never above 50%. These differences imply that *RePBubLik*



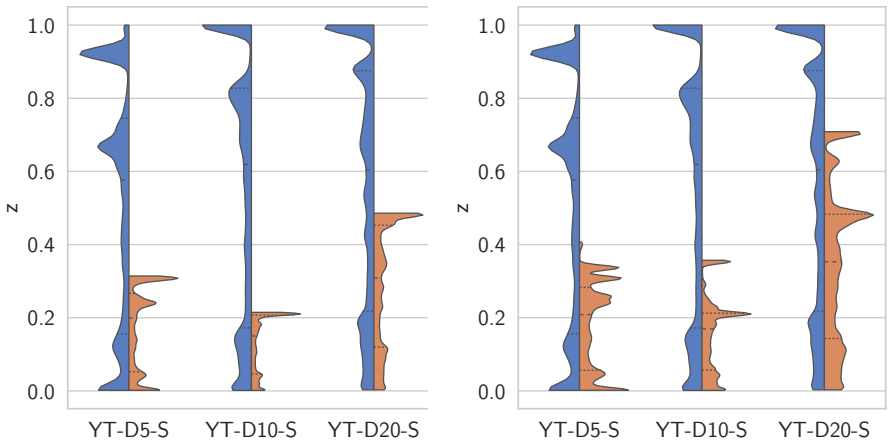
(a) YT-D5-B (b) YT-D5-S

(c) YT-D20-B (d) YT-D20-S

Figure 6.4: Performance of our algorithm (HEU) with varying quality constraints τ.

might not perform well when the in-degree distribution of the graph is not highly skewed.

Robustness w.r.t. threshold of recommendation quality. To investigate the role of the threshold τ of recommendation quality on the output of our algorithm, we test on the YouTube recommendation graphs with the same number of rewiring operations ($k = 50$) but different values of τ in $\{0.5, 0.8, 0.9, 0.99\}$. We present the results in Figure 6.4. As expected, under a more lenient quality constraint ($\tau = 0.5$), the algorithm achieves a larger decease in the value of Z. It is also clear that the differences are



(a) Heuristic (HEU) (b) RePBubLik (RBL)

Figure 6.5: Distribution of the segregation scores (z values) of harmful nodes before (blue) and after (red) performing 50 rewiring operations provided by HEU and RBL.

less evident on graphs with a larger out-degree ($d = 20$). Specifically, for a smaller out-degree ($d = 5$) all the τ configurations except $\tau = 0.5$ tend to stabilize after $k = 20$ rewiring operations. This is because the number of possible rewiring operations constrained by τ is small. It is also evident that the graph size, given different values of τ, does not impact the overall performance of our algorithm.

Total exposure to harmful content. Having tested the effectiveness of our algorithm in reducing the maximum segregation score, we study its effect on the distribution of the segregation scores over all harmful nodes. Figure 6.5 depicts the distribution of the z values before and after the rewiring operations (with $k = 50$ and $\tau = 0.9$) provided by **HEU** and **RBL** on the YouTube recommendation graphs. For each graph, the violin plot in blue (left) denotes the distribution of segregation scores before the rewiring operations and the one in red (right) the distribution after the rewiring operations. The range of segregation scores is normalized

to $[0, 1]$, where the maximum corresponds to the initial segregation. We observe that reducing the maximum segregation also helps reduce the segregation scores of other harmful nodes. Compared to **RBL**, **HEU** generates a distribution more highly concentrated around smaller values; this discrepancy between the distributions is most significant when $d = 20$.

6.6 Summary

In this chapter we studied the problem of reducing the risk of radicalization pathways in *what-to-watch-next* recommenders via edge rewiring on the recommendation graph. We formally defined the segregation score of a radicalized node to measure its potential to trap users into radicalization pathways, and formulated the k-REWIRING problem to minimize the maximum segregation score among all radicalized nodes, while maintaining the quality of the recommendations. We proposed an efficient yet effective greedy algorithm based on the absorbing random walk theory. Our experiments, in the context of video and news recommendations, confirmed the effectiveness of our proposed algorithm. Finally, we showed through empirical evidence how our method, designed to reduce maximum segregation, may actually reduce the *total* segregation generated by all harmful nodes in the graph. However, an analytical reasoning behind this result is missing.

7

Fair and Representative Subset Selection from Data Streams

7.1 Introduction

A crucial task in modern data-driven applications, ranging from influence maximization [KKT03, SHC20] and recommender systems [SQM+17, NTM+18], to nonparametric learning [GK10, BMKK14] and coverage problems [SG09, ELVZ17], is to extract concise summaries from large datasets. In all aforementioned applications, this task is formulated as selecting a subset of items to maximize a utility function that quantifies the "representativeness" (or "utility") of the selected subset. Oftentimes, the objective function satisfies *submodularity*, a property of "diminishing returns" such that adding an item to a smaller set always leads to a larger increase in utility than adding it to a bigger set. Consequently, maximizing submodular set functions subject to a cardinality constraint (i.e., the size of the selected subset is restricted to a given integer k) is general enough to model many practical problems in data mining and machine learning. In this work, we adopt the same formulation for representative item selection.

The classic approach to the cardinality-constrained submodular maximization problem is the GREEDY algorithm proposed by [NWF78], which

achieves an approximation factor of $1 - \frac{1}{e}$ that is NP-hard to improve [Fei98]. In many real-world scenarios, however, the data become too large to fit in memory or arrive incrementally at a high rate – and in such cases, the GREEDY algorithm becomes very inefficient because it requires k repeated sequential scans over the whole dataset. Therefore, streaming algorithms for submodular maximization problems have received much attention recently [AEF$^+$20, GK10, BMKK14, NTM$^+$18, KMZ$^+$19]. Typically, they require only one or a few passes over the dataset, store a very small portion of items in memory, and compute a solution more efficiently than the GREEDY algorithm at the expense of slightly lower quality.

Despite the extensive studies on streaming submodular maximization, unfortunately, it seems that none of the existing methods consider the *fairness* issue of the subsets extracted from data streams. In fact, recent studies [DSB$^+$19, CR20, KMM15, CKS$^+$18] reveal that the data summaries automatically generated by algorithms might be biased with respect to sensitive attributes such as gender, race or ethnicity, and the biases in summaries could be passed to data-driven decision-making processes in education, recruitment, banking, and judiciary systems. Thus, it is necessary to introduce *fairness* constraints into submodular maximization problems so that the selected subset can fairly represent each sensitive attribute in the dataset. Towards this end, we consider that the data stream V comprises l disjoint groups $V_1, V_2, \ldots, V_l$ defined by some sensitive attribute. For example, the groups may correspond to a demographic attribute such as *gender* or *age*. We define the fairness constraint by assigning a cardinality constraint k_i to *each group* V_i with $\sum_{i=1}^{l} k_i = k$. Then, our goal is to maximize the submodular objective function under the constraint that the selected subset contains k_i items from V_i. Note that our fairness constraint can incorporate different concepts of *fairness* by assigning different values of $k_1, k_2, \ldots, k_l$. For example, one can extract a subset that approximately represents the proportion of each group in the dataset by setting $k_i = \frac{|V_i|}{|V|} \cdot k$. As another example, one can also enforce a balanced representation of each group by setting $k_i = \frac{k}{l}$.

Theoretically, the fairness constraint as defined above is a case of

partition matroid constraints [AMT13, JLNN20, KAM19], and thus the optimization problem is reduced to maximizing submodular set functions with matroid constraints. It is not surprising that all existing algorithms for the total cardinality constraint (i.e., the total budget k) cannot be directly used for this problem anymore, because their solutions may not satisfy the fairness constraint (i.e., the group-specific cardinality constraints). Nevertheless, a seminal work by [FNW78] indicates that the GREEDY algorithm with minor modifications is $\frac{1}{2}$-approximate for this problem. However, it still suffers from efficiency issues when processing data streams. In addition, the state-of-the-art streaming algorithms [CGQ15, CK15, FKK18] for matroid-constrained submodular maximization are only $\frac{1}{4}$-approximate and do not provide solutions of the same quality as the GREEDY algorithm efficiently in practice.

In this chapter, we investigate the problem of *streaming submodular maximization with fairness constraints*. Our main contributions are summarized as follows.

- We first formally define the *fair submodular maximization* (FSM) problem and show its NP-hardness. We also describe the $\frac{1}{2}$- approximation GREEDY algorithm for the FSM problem in the offline setting and discuss why it cannot work efficiently in data streams. (Section 7.3)

- We propose a multi-pass streaming algorithm MP-FSM for the FSM problem. Theoretically, MP-FSM requires $O(\frac{1}{\varepsilon} \cdot \log \frac{k}{\varepsilon})$ passes over the dataset, stores $O(k)$ items in memory, and has an approximation ratio of $(\frac{1}{2} - \varepsilon)$ for any constant $\varepsilon > 0$. (Section 7.4.1)

- We further propose a single-pass streaming algorithm SP-FSM for the FSM problem, which requires only one pass over the data stream and offers the same approximation ratio as MP-FSM when an unbounded buffer size is permitted. We also discuss how to adapt SP-FSM heuristically to limit the buffer size to $O(k)$. (Sections 7.4.2 & 7.4.3)

- Finally, we evaluate the performance of our proposed algorithms

against the state-of-the-art baselines in two real-world application scenarios, namely *maximum coverage on large graphs* and *personalized recommendation*. The empirical results on several real-world and synthetic datasets demonstrate the efficiency, effectiveness, and scalability of MP-FSM and SP-FSM. (Section 7.5)

7.2 Related Work

There has been a large body of work on submodular optimization for its wide applications in data mining and machine learning, including influence maximization [KKT03], facility location [LKG+07, LWD16], nonparametric learning [GK10, BMKK14], and group item recommendation [SQM+17]. We refer interested readers to [KG14] for a survey.

The line of research that is the most relevant to this work is *streaming algorithms for submodular maximization*. The seminal work by [NWF78, FNW78] showed that the GREEDY algorithm, which iteratively adds an item that maximally increase the utility with k passes over the dataset, gives approximation ratios of $1 - \frac{1}{e}$ and $\frac{1}{2}$ for maximizing monotone submodular functions with cardinality and matroid constraints, respectively. Then, a series of recent studies [GK10, BMKK14, KMVV15, KMZ+19] proposed multi- or single-pass streaming algorithms for maximizing monotone submodular functions subject to cardinality constraints with the same approximation ratio of $\frac{1}{2} - \varepsilon$. Furthermore, the work of [NTM+18] showed that any single-pass streaming algorithm must use $\Omega(\frac{n}{k})$ memory to achieve an approximation ratio of over $\frac{1}{2}$. They also proposed streaming algorithms with approximation factors better than $\frac{1}{2}$ by assuming that items arrive in random order or running in multiple passes. Also, [AEF+20] proposed a 0.2779-approximation streaming algorithm for maximizing non-monotone submodular functions with cardinality constraints. Moreover, streaming submodular maximization was also studied in different models, e.g., the sliding-window model [ELVZ17, WLT19] where only recent items within a time window are available for selection, the time-decay model [ZSW+19] where the weights of old

items decrease over time, and the deletion-robust model [MKK17, MBN$^+$17, KZK18] where existing items might be removed from the stream. However, all above steaming algorithms are specific for the cardinality constraint and cannot be directly used for the fairness constraint (a case of *matroid*) in this chapter. We note that the *fairness* of submodular maximization problems was also studied in [KZK18]. However, they considered removing sensitive items from the dataset for ensuring fairness, which is different from the problem we consider in this chapter.

A contribution by [CK15] proposed a $\frac{1}{4p}$-approximation single-pass streaming algorithms for maximizing monotone submodular functions with the intersections of p matroid constraints. Similarly, [CGQ15] generalized the algorithm in [CK15] to the case of non-monotone submodular functions. Both algorithms have a $\frac{1}{4}$-approximation for the FSM problem. Moreover, in a recent work by [CHJ$^+$17] improved the approximation ratio for partition matroids to 0.3178 via randomization and relaxation. Also, [FKK18] introduced a subsampling method to accelerate the algorithm of [CGQ15, CK15] while preserving a $\frac{1}{4p}$-approximation ratio (in expectation). Very recently, the work by [HTW20] proposed an $O(\frac{1}{\varepsilon})$-pass $\frac{1}{2+\varepsilon}$-approximation algorithm for monotone submodular maximization with matroid constraints. We implement the aforementioned algorithms from [CGQ15, CK15, FKK18, HTW20] as baselines and compare with them. We do not implement the algorithm in [CHJ$^+$17] since it is not scalable to large-scale data.

Another line of research related to this work is *fair data summarization*. Fair k-center for data summarization was studied in [KAM19, JLNN20, CKR20]. Seminal work was proposed by [CKS$^+$18] proposed a determinantal point process (DPP) based sampling method for fair data summarization. Recently, [DSB$^+$19] considered the fairness issue on summarizing user-generated textual content. Although these studies adopt similar fairness constraints to ours, the proposed methods cannot be applied to the FSM problem since the problems they considered are different from submodular optimization.

7.3 Problem Definition

We consider the problem of selecting a subset of representative items from a dataset V of size n. Our goal is to maximize a non-negative set function $f : 2^V \rightarrow \mathbb{R}^+$, where, for any subset $S \subseteq V$, $f(S)$ quantifies the utility of S, i.e., how well S represents V according to some objective. In many data summarization problems (e.g., [GK10, BMKK14, LWD16, ELVZ17]), the utility function satisfies an intuitive *diminishing returns* property called *submodularity*. To describe it formally, we define the *marginal gain* $\Delta_f(v|S) \coloneqq f(S \cup \{v\}) - f(S)$ as the increase in utility when an item v is added to a set S. A set function f is *submodular* iff $\Delta_f(v|A) \geq \Delta_f(v|B)$ for any $A \subseteq B \subseteq V$ and $v \in V \setminus B$. This means that adding an item e to a set A leads to at least as much utility gain as adding v to a superset B of A. Additionally, a submodular function f is *monotone* iff $\Delta_f(v|S) \geq 0$ for any $S \subseteq V$ and $v \in V \setminus S$, i.e., adding a new item v will not decrease the utility of S. In this work, we assume that the function f is both monotone and submodular. Moreover, following most existing works [GK10, BMKK14, LKG+07, ELVZ17, CGQ15, FKK18, KMZ+19, NTM+18], we assume that the utility $f(S)$ of any set $S \subseteq V$ is given by a value oracle – i.e., the value of $f(S)$ is retrieved in constant time.

Let us consider the following canonical optimization problem: given a monotone submodular set function f and a dataset V, find a subset of size k from V that maximizes the function f, i.e.,

$$\max_{S \subseteq V} f(S) \quad \text{s.t.} \quad |S| = k \tag{7.1}$$

The problem in Eq. 7.1 is referred to as the cardinality-constrained submodular maximization (CSM) problem and proven to be NP-hard [Fei98] for many classes of submodular functions. And the well-known greedy algorithm of [NWF78] achieves a $(1-\frac{1}{e})$-approximation for this problem.

Now let us introduce *fairness* into the CSM problem. Suppose that the dataset V is partitioned into l (disjoint) groups, each of which corresponds to a sensitive class, and V_i is the set of items from the i-th group in V with $\bigcup_{i=1}^{l} V_i = V$. Then, for each group, we demand that the solution S must

contain k_i items from V_i, with $\sum_{i=1}^{l} k_i = k$. Formally, we define the fair submodular maximization (FSM) problem as follows:

$$S^* = \arg\max_{S \subseteq V} f(S) \quad \text{s.t.} \quad |S \cap V_i| = k_i, \forall i \in [l] \tag{7.2}$$

where S^* and $\texttt{OPT} = f(S^*)$ denote the optimal solution and its utility. The values of $k_1, \ldots, k_l \in \mathbb{Z}^+$ are given as input to the problem (here, we assume $k_i > 0$ since we can simply ignore all items in V_i if $k_i = 0$) and determined according to the notion of fairness. For example, one can use $k_i = \frac{n_i}{n} \cdot k$ where $n_i = |V_i|$ to obtain a *proportional representation*. As another example, an *equal representation* can be acquired by setting $k_i = \frac{k}{l}$ for all $i \in [l]$.

Algorithm 3: GREEDY

Input : Dataset V, groups $V_1, \ldots, V_l \subseteq V$, size constraint $k \in \mathbb{Z}^+$, group size constraints $k_1, \ldots, k_l \in \mathbb{Z}^+$
Output: Solution S for the FSM problem on V

```
Initialize the solution S ← ∅;
for j ← 1, ..., k do
    for i ← 1, ..., l do
        if |S ∩ V_i| < k_i then
            Pick an item v_i^* ← arg max_{v ∈ V_i ∩ V} Δ_f(v|S);
        else
            v_i^* ← NULL;
    Select an item v^* ← arg max_{i ∈ [l] : v_i^* ≠ NULL} Δ_f(v_i^*|S);
    S ← S ∪ {v^*}, V ← V \ {v^*};
return S;
```

The FSM problem in Eq. 7.2 is still NP-hard because the CSM problem in Eq. 7.1 is its special case when $l = 1$. Nevertheless, a generalized GREEDY algorithm first proposed in [FNW78] provides a $\frac{1}{2}$-approximate solution for the FSM problem, since the fairness constraint we consider is a case of the *partition matroid* constraint. The procedure of GREEDY is

described in Algorithm 3. Starting from $S = \varnothing$, it iteratively adds an item v^* with the maximum utility gain $\Delta_f(v^*|S)$ to the current solution S. To guarantee that solution S satisfies the fairness constraint, it excludes from consideration all items of V_i once there are k_i items from V_i in S, i.e., $|S \cap V_i| = k_i$. The solution S after k iterations is returned for the FSM problem. The running time of GREEDY is $O(nk)$ because it requires k passes through the dataset and evaluates the value of f at most n times per pass for identifying v_i^*. Therefore, GREEDY becomes very inefficient when the dataset size is large; even worse, GREEDY cannot work in the single-pass streaming setting if the dataset does not fit in the memory. In what follows, we investigate the FSM problem in streaming settings.

7.4 Our Algorithms

In this section, we present our proposed algorithms for the fair submodular maximization (FSM) problem in data streams. Firstly, we propose a multi-pass streaming algorithm called MP-FSM. For any constant $\varepsilon \in (0, 1)$, MP-FSM requires $O\big(\frac{1}{\varepsilon} \cdot \log \frac{k}{\varepsilon}\big)$ passes over the dataset, stores $O(k)$ items in memory, and provides a $\frac{1}{2}(1 - \varepsilon)$-approximate solution for the FSM problem. Secondly, we propose a single-pass streaming algorithm called SP-FSM on the top of MP-FSM. SP-FSM has an approximation ratio of $\frac{1}{2} - \varepsilon$ and sublinear update time per item. But it might keep $O(n)$ items in a buffer for post-processing in the worst case, and thus its space complexity is $O(n)$. Therefore, we further discuss how to bound the buffer size of SP-FSM when the memory space is limited and how the approximation ratio of SP-FSM is affected accordingly.

7.4.1 The Multi-Pass Streaming Algorithm

In this subsection, we present our multi-pass streaming algorithm called MP-FSM for the FSM problem. In general, MP-FSM adopts a threshold-based approach similar to existing streaming algorithms for the CSM problem [BMKK14, KMVV15, NTM$^+$18, KMZ$^+$19]. The high-level

idea of the threshold-based approach is to process items in a data stream sequentially with a threshold τ: for each item v received from the stream, it will accept v into a solution S if $\Delta_f(v|S)$ reaches τ and discard v otherwise. But differently from most thresholding algorithms [BMKK14, KMVV15, KMZ+19] for the CSM problem, which run in only one pass and use a fixed threshold for each candidate solution, MP-FSM scans the dataset in multiple passes using a decreasing threshold to determine whether to include an item in each pass so that the solution has a constant approximation ratio while satisfying the fairness constraint.

We present the detailed procedure of MP-FSM in Algorithm 4. In the first pass, it finds the item v_{max} with the maximum utility $\delta_{max} = f(\{v_{max}\})$ among all items in the dataset V. The purpose of finding v_{max} is to determine the range of thresholds to be used in subsequent passes. Meanwhile, it keeps a random sample R_i of k_i items uniformly from V_i for each $i \in [l]$, which will be used for post-processing to guarantee that the solution satisfies the fairness constraint. Then, it initializes a solution S containing only v_{max} and a threshold $\tau = (1-\varepsilon) \cdot \delta_{max}$ for the second pass. After that, it scans the dataset V sequentially in multiple passes. In each pass, it decreases the threshold τ by $(1-\varepsilon)$ times and adds an item $v \in V_i$ to the current solution S if the marginal gain of v w.r.t. S reaches τ and there are fewer than k_i items in S from V_i. When the solution S has contained k items or the threshold τ has been decreased to be lower than $\frac{\varepsilon}{k} \cdot \delta_{max}$, no more passes are needed. Finally, if the solution S does not satisfy the fairness constraint, it will add items from random samples to S for ensuring its validity.

Next, we provide some theoretical analysis for the MP-FSM algorithm. First, we provide the approximation ratio of MP-FSM in Theorem 2; and then, the complexity of MP-FSM in Theorem 3.

Theorem 2. *For any parameter $\varepsilon \in (0,1)$,* MP-FSM *in Algorithm 4 is a $\frac{1}{2}(1-\varepsilon)$-approximation algorithm for the FSM problem.*

Proof. Let O be the optimal solution for the FSM problem on dataset V and $O_i = O \cap V_i$ be the intersection of O and V_i for each $i \in [l]$. We consider that MP-FSM runs in p passes and $S^{(j)}$ $(1 \leq j \leq p)$ is the

Algorithm 4: MP-FSM

Input : Dataset V, groups $V_1, \ldots, V_l \subseteq V$, size constraint $k \in \mathbb{Z}^+$, group size constraints $k_1, \ldots, k_l \in \mathbb{Z}^+$, parameter $\varepsilon \in (0, 1)$

Output: Solution S for the FSM problem on V

`/* Pass 1: Get` v_{max} `and reservoir sampling */`

$v_{max} \leftarrow \arg\max_{v \in V} f(\{v\})$ and $\delta_{max} \leftarrow f(\{v_{max}\})$;

Keep a random sample R_i of k_i items uniformly from V_i for each $i \in [l]$ via reservoir sampling [Vit85];

`/* Pass 2 to` p`: Compute solution` S `*/`

$S \leftarrow \{v_{max}\}$ and $\tau \leftarrow (1 - \varepsilon) \cdot \delta_{max}$;

while $\tau > \frac{\varepsilon}{k} \cdot \delta_{max}$ **do**

  foreach *item* $v \in V \setminus S$ **do**

    if $v \in V_i$ *and* $|S \cap V_i| < k_i$ *and* $\Delta_f(v|S) \geq \tau$ **then**

      $S \leftarrow S \cup \{v\}$;

  if $|S| = k$ **then**

    break;

  else

    $\tau \leftarrow (1 - \varepsilon) \cdot \tau$;

`/* Post processing: Ensure fairness */`

while $\exists i \in [l] : |S \cap V_i| < k_i$ **do**

  Add items in R_i to S until $|S \cap V_i| = k_i$;

return S;

partial solution of MP-FSM after j passes. For any subset O_i of O and the solution $S^{(p)}$ after p passes, we have either (1) $|S^{(p)} \cap V_i| = k_i$ or (2) $|S^{(p)} \cap V_i| < k_i$. If $|S^{(p)} \cap V_i| = k_i$, there are two cases for each item $o \in O_i$: (1.1) $o \in S^{(p)}$ and (1.2) $o \notin S^{(p)}$. In Case (1.1), we have $\Delta_f(o|S^{(p)}) = 0$. In Case (1.2), we compare o with an item s from V_i added to the solution during the j-th pass. Since both o and s cannot be added in the $(j-1)$-th pass and $|S^{(j-1)} \cap V_i| < k_i$, it is safe to say that the marginal gains of o and s w.r.t. $S^{(j-1)}$ do not reach the threshold $\tau^{(j-1)}$ of the $(j-1)$-th pass. As s is added in the j-th pass, we have $\Delta_f(s|S') \geq \tau^{(j)}$ where $S' \subseteq S^{(j)}$ is the partial solution before s is added. Therefore, we have the following sequence of inequalities:

$$\Delta_f(o|S^{(p)}) \leq \Delta_f(o|S^{(j-1)}) < \tau^{(j-1)} = \frac{\tau^{(j)}}{1-\varepsilon} \leq \frac{\Delta_f(s|S')}{1-\varepsilon} \tag{7.3}$$

Then, if $|S^{(p)} \cap V_i| < k_i$, there are also two cases for $o \in O_i$: (2.1) $o \in S^{(p)}$ and (2.2) $o \notin S^{(p)}$. Case (2.1) is exactly the same as Case (1.1). In Case (2.2), we have:

$$\Delta_f(o|S^{(p)}) < \tau^{(p)} \leq \frac{\varepsilon}{k(1-\varepsilon)} \cdot \delta_{max} \tag{7.4}$$

where $\tau^{(p)}$ is the threshold of the p-th pass.

Next, we divide O into two disjoint subsets O' and O'' as follows: $O' = \cup_{i'} O_{i'}$ where $|S^{(p)} \cap V_{i'}| = k_{i'}$, i.e., all items from groups satisfying Case (1), and $O'' = O \setminus O'$, i.e., all items from groups satisfying Case (2). We define an injection $\pi : O' \rightarrow S^{(p)}$ that maps each item in O' to an item in $S^{(p)}$ as follows: If $o \in S^{(p)}$, then $\pi(o) = o$; otherwise, $\pi(o)$ will be an arbitrary item $s \in S^{(p)}$ from the same group as o and $s \notin O$. Based on the result of Eq. 7.3, we can get the following inequalities for O':

$$\sum_{o \in O'} \Delta_f(o|S^{(p)}) \leq \frac{\sum_{\pi(o) \in S^{(p)}} \Delta_f(\pi(o)|S')}{1-\varepsilon} \leq \frac{f(S^{(p)})}{1-\varepsilon} \tag{7.5}$$

Here, S' denotes the partial solution before $\pi(o)$ is added and the second inequality is acquired from the fact that $f(S^{(p)}) = \sum_{s \in S^{(p)}} \Delta_f(s|S')$.

Then, based on the result of Eq. 7.4, we have the following inequalities for O'':

$$\sum_{o \in O''} \Delta_f(o|S^{(p)}) \leq \frac{\varepsilon \cdot |O''|}{k(1-\varepsilon)} \cdot \delta_{max} \leq \frac{\varepsilon}{1-\varepsilon} \cdot f(S^{(p)}) \tag{7.6}$$

because $|O''| < k$ and $\delta_{max} \leq f(S^{(p)})$. Finally, we have the following sequence of inequalities from Eq. 7.5 and 7.6:

$$\begin{aligned} f(O \cup S^{(p)}) - f(S^{(p)}) &= \sum_{o \in O'} \Delta_f(o|S^{(p)}) + \sum_{o \in O''} \Delta_f(o|S^{(p)}) \\ &\leq \frac{1}{1-\varepsilon} \cdot f(S^{(p)}) + \frac{\varepsilon}{1-\varepsilon} \cdot f(S^{(p)}) = \frac{1+\varepsilon}{1-\varepsilon} \cdot f(S^{(p)}) \end{aligned}$$

Since $\texttt{OPT} = f(O) \leq f(O \cup S^{(p)})$, we have $\texttt{OPT} \leq f(S^{(p)}) + \frac{1+\varepsilon}{1-\varepsilon} \cdot f(S^{(p)}) \leq \frac{2}{1-\varepsilon} \cdot f(S^{(p)})$. Finally, we conclude the proof from the fact that $f(S) \geq f(S^{(p)}) \geq \frac{1}{2}(1-\varepsilon) \cdot \texttt{OPT}$. □

Theorem 3. MP-FSM *in Algorithm 4 requires* $O\big(\frac{1}{\varepsilon} \cdot \log \frac{k}{\varepsilon}\big)$ *passes over the dataset* V*, stores at most* $O(k)$ *items, and has* $O\big(\frac{n}{\varepsilon} \cdot \log \frac{k}{\varepsilon}\big)$ *time complexity.*

Proof. First of all, since the threshold τ is decreased by $1-\varepsilon$ times after one pass, $\tau^{(2)} = (1-\varepsilon) \cdot \delta_{max}$, and $\tau^{(p)} \geq \frac{\varepsilon}{k} \cdot \delta_{max}$, we get $(1-\varepsilon)^{p-1} \geq \frac{\varepsilon}{k}$. Taking the logarithm on both sides of the last inequality and the Taylor expansion of $\log(1-\varepsilon)$, we have $p \leq 1 + \frac{1}{\log(1-\varepsilon)} \cdot \log \frac{\varepsilon}{k} \leq 1 + \frac{1}{\varepsilon} \cdot \log \frac{k}{\varepsilon}$ and thus the number p of passes in MP-FSM is $O\big(\frac{1}{\varepsilon} \cdot \log \frac{k}{\varepsilon}\big)$. Furthermore, MP-FSM only stores items in the solution and random samples for post-processing, both of which contain at most k items. Hence, MP-FSM stores at most $O(k)$ items. Finally, because MP-FSM evaluates the value of function f at most n times per pass, the total number of function evaluations is $O\big(\frac{n}{\varepsilon} \cdot \log \frac{k}{\varepsilon}\big)$. □

7.4.2 The Single-Pass Streaming Algorithm

In this subsection, we present our single-pass streaming algorithm called SP-FSM for the FSM problem. Generally, SP-FSM is based on a threshold-

based approach, similar to MP-FSM. However, several adaptations are required so that SP-FSM can provide an approximate solution in only one pass over the dataset. First of all, because v_{max} and δ_{max} are unknown in advance, SP-FSM should keep track of them from received items, dynamically decide a sequence of thresholds based on the observed δ_{max}, and maintain a candidate solution for each threshold (instead of keeping only one solution over multiple passes in MP-FSM). Furthermore, as only one pass is permitted, an item will be unrecoverable once it is discarded. To provide a theoretical guarantee for the quality of solutions in adversarial settings, SP-FSM keeps a buffer to store items that are neither included into solutions nor safely discarded. Finally, whenever a solution is requested during the stream, OP-RSM will reconsider the buffered items for post-processing by attempting to add them greedily to candidate solutions. We will show that SP-FSM has an approximation ratio of $\frac{1}{2} - \varepsilon$ with a judicious choice of parameters when the buffer size is unlimited.

The detailed procedure of SP-FSM is presented in Algorithm 5. Here, δ_{max} keeps the maximum utility of any single item among all items received so far, $\texttt{LB}$ maintains the lower bound of $\texttt{OPT}$ estimated from candidate solutions, B stores the buffered items, and R_i is a set of k_i items sampled uniformly from all received items in V_i. In addition, two parameters α and β are used to control the number of candidate solutions and the number of buffered items, respectively. Generally, larger α means bigger gaps between neighboring thresholds and thus fewer candidates, while larger β means more rigorous conditions for adding an item to the buffer and naturally smaller buffer sizes. The procedure for stream processing of SP-FSM is given in Line 2–14. For each item $v \in V_i$ received from V, it first updates the value of δ_{max} and the sample R_i w.r.t. v accordingly. Then, it maintains a sequence T of thresholds picked from a geometric progression $\{(1+\alpha)^j | j \in \mathbb{Z}\}$ and a candidate solution S_τ for each $\tau \in T$. Specifically, the upper bound of the threshold is set to δ_{max} since $S_\tau = \varnothing$ for any $\tau > \delta_{max}$; the lower bound of the threshold is set to $\frac{\max\{\delta_{max}, \texttt{LB}\}}{2k}$ because any candidate with a threshold lower than $\frac{\texttt{OPT}}{2k}$ is safe to be discarded (as shown in our theoretical analysis later) and $\max\{\delta_{max}, \texttt{LB}\}$ is the lower bound of $\texttt{OPT}$. After maintaining the thresholds and their corre-

Algorithm 5: SP-FSM

Input : Data stream V, groups $V_1, \ldots, V_l \subseteq V$, size constraint $k \in \mathbb{Z}^+$, group size constraints $k_1, \ldots, k_l \in \mathbb{Z}^+$, parameters $\alpha, \beta \in (0, 1)$

Output: Solution S for the FSM problem on V

$\delta_{max} \leftarrow 0$, $\texttt{LB} \leftarrow 0$, $B \leftarrow \varnothing$, and $R_i \leftarrow \varnothing$ for each $i \in [l]$;

/* Stream processing */

foreach *item* $v \in V_i$ *received from* V **do**

  $\delta_{max} \leftarrow \max\{\delta_{max}, f(\{v\})\}$;

  Update R_i w.r.t. v using reservoir sampling [Vit85];

  $T \leftarrow \{(1+\alpha)^j | j \in \mathbb{Z}, \frac{\max\{\delta_{max}, \texttt{LB}\}}{2k} \leq (1+\alpha)^j \leq \delta_{max}\}$;

  Discard S_τ for all $\tau \notin T$;

  Initialize $S_\tau \leftarrow \varnothing$ for each τ newly added to T;

  foreach $\tau \in T$ **do**

    if $|S_\tau \cap V_i| < k_i$ **then**

      if $\Delta_f(v|S_\tau) \geq \tau$ **then**

        $S_\tau \leftarrow S_\tau \cup \{v\}$;

      else if $\Delta_f(v|S_\tau) \geq \frac{\beta \cdot \texttt{LB}}{k}$ **then**

        $B \leftarrow B \cup \{v\}$;

  $\texttt{LB} \leftarrow \max_{\tau \in T} f(S_\tau)$;

/* Post processing */

Let τ' be the smallest $\tau \in T$ such that $|S_\tau \cap V_i| < k_i$ for each $i \in [l]$ or the largest $\tau \in T$ if there exists some i such that $|S_\tau \cap V_i| = k_i$ for every S_τ;

foreach $\tau \leq \tau'$ *in* T **do**

  Run GREEDY in Algorithm 3 to add items from buffer B and samples R_i ($i \in [l]$) to S_τ until $|S_\tau| = k$;

return $S \leftarrow \arg\max_{\tau \in T} f(S_\tau)$;

sponding candidates, SP-FSM evaluates the marginal gain $\Delta_f(v|S_\tau)$ of v for each candidate S_τ with threshold $\tau \in T$: Similar to MP-FSM, it will add v to S_τ if $\Delta_f(v|S_\tau)$ reaches τ and $|S_\tau \cap V_i| < k_i$; Additionally, it will add v to the buffer B if $\Delta_f(v|S_\tau)$ is at least $\frac{\beta \cdot \mathtt{LB}}{k}$ but less than τ. Finally, $\mathtt{LB}$ is updated to the utility of the best solution found so far. The procedure for post-processing of SP-FSM is shown in Lines 15–17. It first finds out the smallest $\tau \in T$ such that $|S_\tau \cap V_i| < k_i$ for each $i \in [l]$ as τ'; if such τ does not exist, i.e., there exists some i such that $|S_\tau \cap V_i| = k_i$ for every S_τ, the largest $\tau \in T$ is used as τ'. For each $\tau \leq \tau'$ in T, it runs GREEDY in Algorithm 3 to reevaluate the items in B and R_i ($i \in [l]$) and add them to S_τ until $|S_\tau| = k$. Lastly, the candidate solution with the maximum utility after post-processing is returned as the final solution.

Next, we will provide the theoretical analysis for the SP-FSM algorithm. First, in Lemma 4, we analyze the special cases when the solution returned after stream processing (without post-processing) can achieve a good approximation ratio.

Lemma 4. *Assume that* $\frac{\mathtt{OPT}}{2k} \leq \tau \leq \frac{(1+\alpha)\cdot\mathtt{OPT}}{2k}$. *If either* $|S_\tau| = k$ *or* $|S_\tau \cap V_i| < k_i$ *for all* $i \in [l]$, *then* $f(S_\tau) \geq \frac{1-\alpha}{2} \cdot \mathtt{OPT}$.

Proof. First of all, when $|S_\tau| = k$, it holds that $f(S_\tau) \geq k\tau \geq k \cdot \frac{\mathtt{OPT}}{2k} = \frac{1}{2} \cdot \mathtt{OPT} \geq \frac{1-\alpha}{2} \cdot \mathtt{OPT}$. Then, when $|S_\tau \cap V_i| < k_i$ for all $i \in [l]$, we have $\Delta_f(v|S_\tau) < \tau$ for any $v \in V \setminus S_\tau$. Let O be the optimal solution for the FSM problem on V. We can acquire that

$$\begin{aligned} f(O \cup S_\tau) - f(S_\tau) &\leq \sum_{o \in O \setminus S_\tau} \Delta_f(o|S_\tau) < k\tau \\ &\leq k \cdot \frac{(1+\alpha)\cdot\mathtt{OPT}}{2k} = (1+\alpha)\cdot\frac{\mathtt{OPT}}{2} \end{aligned}$$

Therefore, we have $f(S_\tau) \geq f(O \cup S_\tau) - (1+\alpha)\cdot\frac{\mathtt{OPT}}{2} \geq \mathtt{OPT} - (1+\alpha)\cdot\frac{\mathtt{OPT}}{2} = \frac{1-\alpha}{2}\cdot\mathtt{OPT}$. Finally, we conclude the proof by considering both cases collectively. □

Lemma 4 is useful because one of the thresholds $\tau \in T$ of SP-FSM (Line 5 of Algorithm 5) must satisfy the first condition $\frac{\mathtt{OPT}}{2k} \leq \tau \leq$

$\frac{(1+\alpha)\cdot \texttt{OPT}}{2k}$ of the lemma. This is because T is a geometric progression with a scale factor of $(1+\alpha)$ and spans the range $[\frac{\max\{\delta_{max}, \texttt{LB}\}}{2k}, \delta_{max}]$, with $\max\{\delta_{max}, \texttt{LB}\} \leq \texttt{OPT} \leq k \cdot \delta_{max}$.

This implies that, if the remaining conditions of Lemma 4 were satisfied as well, the solution of SP-FSM after stream processing would have the strong approximation guarantee given by Lemma 4. Intuitively, this would be the case when the utility distribution of items was generally "balanced" among groups, so that either all or none of the group budgets would be exhausted by the end of the stream processing procedure. However, in case that the utilities are highly imbalanced among groups, the approximation ratio would become significantly lower. On the one hand, SP-FSM might miss high-utility items in some groups from the stream because the threshold is too low and the solution has been filled by earlier items with lower utilities in these groups. On the other hand, SP-FSM might not include enough items from the other groups because the threshold is too high for them. Note that, for $\frac{\texttt{OPT}}{2k} \leq \tau \leq \frac{(1+\alpha)\cdot \texttt{OPT}}{2k}$, Lemma 4 allows the approximation factor of S_τ to drop to $\frac{\min_{i\in[l]} k_i \tau}{\texttt{OPT}} \geq \min_{i\in[l]} \frac{k_i}{2k} \geq \frac{1}{2k}$ when some group budgets are exhausted but the others are not.

Therefore, we further include the buffer and post-processing procedures in SP-FSM so that it still achieves a constant approximation independent of k for an arbitrary group size constraint. In Lemma 5, we analyze the approximation ratio of the solution returned by SP-FSM after post-processing.

Lemma 5. *Let τ' be chosen according to Line 15 of Algorithm 5. It holds that $f(S_{\tau'}) \geq \frac{1-\beta}{2+\alpha} \cdot \texttt{OPT}$ after post-processing.*

Proof. We consider two cases separately: (1) $|S_{\tau'} \cap V_i| < k_i$ for each $i \in [l]$ or (2) τ' is the maximum in T. In Case (1), we divide the items in the optimal solution O into three disjoint subsets: $O_1 = O \cap S_{\tau'}$, i.e., items included in $S_{\tau'}$ during stream and post processing; $O_2 = O \cap (B \setminus S_{\tau'})$, i.e., items stored in the buffer but not added to $S_{\tau'}$; $O_3 = O \cap (V \setminus (B \cup S_{\tau'}))$, i.e., items discarded during stream processing. For each $o \in O_2$, we can always find an item $s \in S_{\tau'}$ from the same group as o such that $\Delta_f(s|S') \geq \Delta_f(o|S') \geq \Delta_f(o|S_{\tau'})$where $S' \subseteq S_{\tau'}$ is the partial solution

when s is added. This is because GREEDY always picks the item with the maximum marginal gain within each group. In addition, for each $o \in O_3$, we have $\Delta_f(o|S_{\tau'}) \leq \frac{\beta \cdot \texttt{LB}}{k} \leq \frac{\beta \cdot \texttt{OPT}}{k}$. Therefore, we have

$$\begin{aligned} f(O \cup S_{\tau'}) - f(S_{\tau'}) &\leq \sum_{o \in O \setminus S_{\tau'}} \Delta_f(o|S_{\tau'}) \\ &= \sum_{o \in O_2} \Delta_f(o|S_{\tau'}) + \sum_{o \in O_3} \Delta_f(o|S_{\tau'}) \\ &\leq \sum_{s \in S_{\tau'}} \Delta_f(s|S') + \beta \cdot \texttt{OPT} \\ &= f(S_{\tau'}) + \beta \cdot \texttt{OPT} \end{aligned}$$

where S' is the partial solution when s is added to $S_{\tau'}$. And we conclude that $f(S_{\tau'}) \geq \frac{1-\beta}{2} \cdot \texttt{OPT}$ from the above inequalities. In Case (2), we have τ' is the maximum in T and thus $\tau' \in [\frac{\delta_{max}}{1+\alpha}, \delta_{max}]$. We divide O into O_1, O_2, O_3 in the same way as Case (1). It is easy to see that the results for O_1 and O_3 are exactly the same as Case (1). The only difference is that there may exist some items in O_2 rejected by $S_{\tau'}$ because their groups have been filled in $S_{\tau'}$. For any $o \in O_2$, we have $\Delta_f(o|S_{\tau'}) \leq \delta_{max} \leq (1+\alpha) \cdot \tau' \leq (1+\alpha) \cdot \Delta_f(s|S')$ where s is from the same group as o and S' is the partial solution when s is added. Accordingly, we can get $\texttt{OPT} - f(S_{\tau'}) \leq (1+\alpha) \cdot f(S_{\tau'}) + \beta \cdot \texttt{OPT}$ and thus $f(S_{\tau'}) \geq \frac{1-\beta}{2+\alpha} \cdot \texttt{OPT}$ in both cases. □

Next, we give the approximation ratio and complexity of SP-FSM in Theorems 6 and 7, respectively.

Theorem 6. *Assuming that $\alpha, \beta = O(\varepsilon)$, SP-FSM in Algorithm 5 is a $(\frac{1}{2} - \varepsilon)$-approximation algorithm for the FSM problem.*

Proof. According to the results of Lemmas 4 and 5, we have $f(S) \geq \frac{1-\beta}{2+\alpha} \cdot \texttt{OPT}$ for the solution S returned by Algorithm 5. By assuming $\alpha, \beta = O(\varepsilon)$, we conclude the proof. □

Theorem 7. *Assuming that $\alpha, \beta = O(\varepsilon)$,* SP-FSM *in Algorithm 5 requires one pass over the data stream V, stores at most $O\big(\frac{k \log k}{\varepsilon} + |B|\big)$ items, has*

$O\big(\frac{\log k}{\varepsilon}\big)$ *update time per item for stream processing, and takes* $O\big(\frac{k \log k}{\varepsilon} \cdot (|B| + k)\big)$ *time for post-processing.*

Proof. The number $|T|$ of thresholds maintained at any time satisfies that $(1+\alpha)^{|T|} \leq 2k$. Using the Taylor expansion of $\log(1+\alpha)$, we have $|T| \leq \frac{\log 2k}{\log(1+\alpha)} \leq \frac{\log 2k}{\alpha} = O\big(\frac{\log k}{\alpha}\big)$. Therefore, the number of function evaluations per item is $O\big(\frac{\log k}{\varepsilon}\big)$. Since each candidate solution contains at most k items, the total number of items stored in SP-FSM is $O\big(\frac{k \log k}{\varepsilon} + |B|\big)$. For each candidate solution S_τ, the post-processing procedure runs in $(k - |S_\tau|)$ iterations and processes at most $(|B| + k)$ items at each iteration. Therefore, it takes $O\big(\frac{k \log k}{\varepsilon} \cdot (|B| + k)\big)$ time for post-processing. □

7.4.3 SP-FSM with Bounded Buffer Size

From the above results, we can see that SP-FSM may store $O(n)$ items in the buffer and take $O\big(\frac{nk \log k}{\varepsilon}\big)$ time for post-processing in the worst case. In practice, a streaming algorithm is often required to process massive data streams with limited time and memory (sublinear to or independent of n). And it is not favorable for SP-FSM to store an unlimited number of items in the buffer B. Therefore, we propose a simple strategy for SP-FSM to manage the buffered items so that the buffer size is always bounded at the expense of lower approximation ratios in adversary settings.

We consider that the maximum buffer size is restricted to $k' = O(k)$ and extra items should be dropped from B once its size exceeds k'. The following rules are considered for buffer management. Firstly, since `LB` increases over time, it is safe to drop at any time during stream processing any item already in the buffer whose marginal gain is lower than $\frac{\beta \cdot \texttt{LB}}{k}$ for the current value of `LB`, without affecting the theoretical guarantee. Secondly, to avoid duplications, if an item is added to some candidate solution but needs to be buffered for another, it is not necessary to add this item to the buffer because the algorithm has already stored this item. In this case, items in both candidates and the buffer should be used for post-

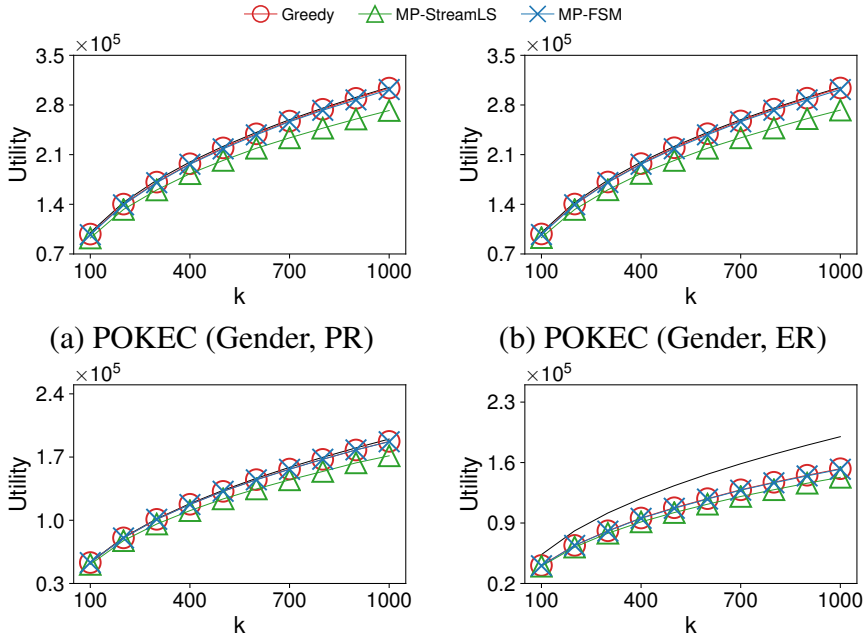
processing. Thirdly, as the buffer is used for storing high-utility items for post-processing, the items with larger marginal gains should have higher priorities to be stored. If the buffer size still exceeds k' after (safely) dropping items using the first two rules, it is required to sort the items in B in a descending order of marginal gain $\delta(v) = \max_{\tau \in T} \Delta_f(v, S_\tau)$ and drop the item v with the lowest $\delta(v)$ until $|B| = k'$. Fourthly, considering the fairness constraint, it will not drop any item v from V_i anymore if $|B \cap V_i| \leq k_i$ even if $\delta(v)$ is among the lowest marginal gains. In this case, it will drop the item with the lowest $\delta(v)$ from V_i with $|B \cap V_i| > k_i$ instead.

The first two rules above have no effect on the theoretical guarantee on the approximation ratio of SP-FSM. The latter two rules will lower the approximation ratio of SP-FSM in some cases. Let v' be the item with the largest $\delta(v)$ among all items dropped due to Rule (3) or (4). The approximation ratio of SP-FSM will drop to $\frac{1-\beta'}{2}$ where $\beta' = \frac{k \cdot \delta(v')}{\texttt{LB}}$. Once $\beta' \geq 1 - \frac{1}{k}$, the approximation ratio will become $\frac{1}{2k}$ in the worst case. Nevertheless, according to our experimental results in Section 7.5, SP-FSM provides high-quality solutions empirically with very small buffer sizes (i.e., $k' = 2k$).

7.5 Experiments

The aim of our experiments is three-fold. First, we aim to quantify "the prices of fairness and streaming", i.e., the loss in solution utility caused by introducing fairness constraints and restricting data access to the streaming model. Second, we aim to demonstrate the improvements of MP-FSM upon existing algorithms in the multi-pass streaming setting. Third, we aim to illustrate that SP-FSM (with unlimited/bounded buffer sizes) outperforms existing single-pass streaming algorithms.

Towards this end, we perform extensive experiments on two applications, namely *maximum coverage on large graphs* and *personalized recommendation*, for evaluation. We compare MP-FSM with the following two multi-pass baselines:



(a) POKEC (Gender, PR)

(b) POKEC (Gender, ER)

(c) POKEC (Age, PR)

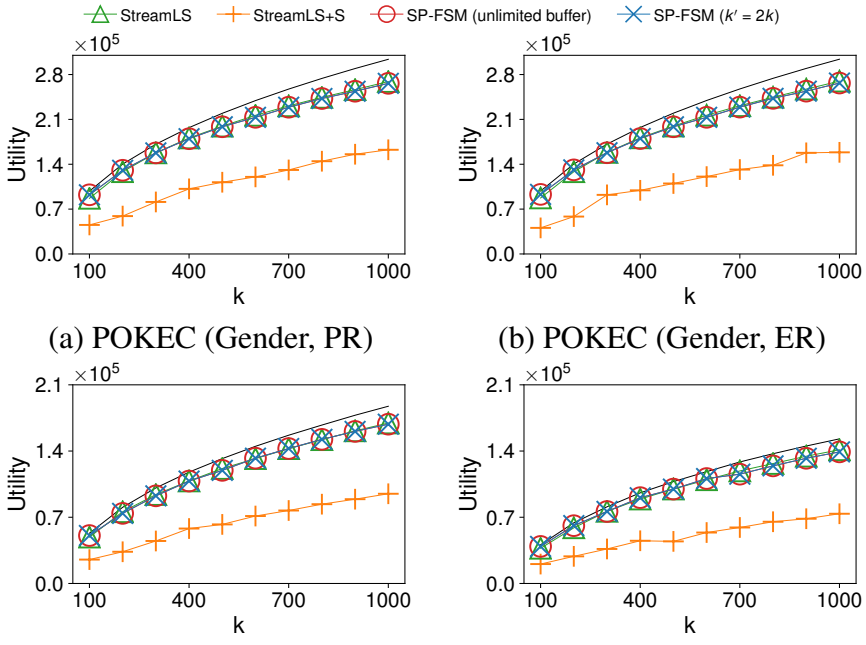
(d) POKEC (Age, ER)

Figure 7.1: Solution utilities of multi-pass algorithms on POKEC. The solution utilities of GREEDY without fairness constraints are plotted as black lines to demonstrate "the price of fairness".

- GREEDY [FNW78]: a $\frac{1}{2}$-approximation k-pass greedy algorithm.
- MP-STREAMLS [HTW20]: a $\frac{1}{2+\varepsilon}$-approximation $O(\frac{1}{\varepsilon})$-pass streaming algorithm.

Moreover, we compare SP-FSM with the following two single-pass baselines:

- STREAMLS [CK15, CGQ15]: a $\frac{1}{4}$-approximation streaming algorithm.
- STREAMLS+S [FKK18]: an improved version of STREAMLS with subsampling. The subsampling rate q is set to 0.1.



(a) POKEC (Gender, PR)

(b) POKEC (Gender, ER)

(c) POKEC (Age, PR)

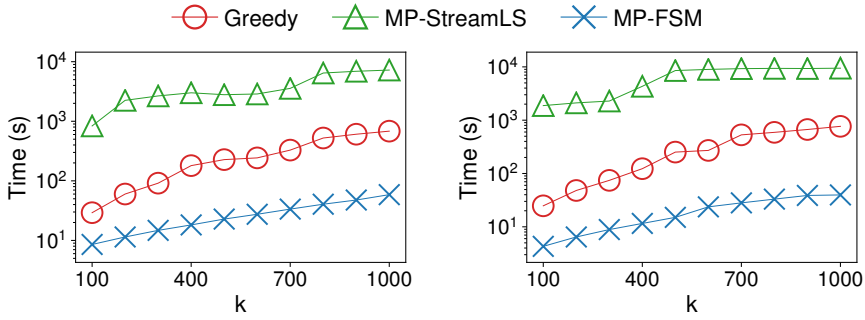
(d) POKEC (Age, ER)

Figure 7.2: Solution utilities of single-pass algorithms on POKEC. The solution utilities of GREEDY are plotted as black lines to show "the price of streaming".

All algorithms were implemented in Python 3.6, and the experiments were conducted on a server running Ubuntu 16.04 with an Intel Broadwell 2.40GHz CPU and 29GB memory. For each of the experiments we invoked our algorithms with the following parameter values: MP-FSM with $\varepsilon = 0.2$; SP-FSM with $\alpha, \beta = 0.5$ and buffer size $k' = 2k$ for those cases where the buffer size is bounded.

7.5.1 Maximum Coverage on Large Graphs

Maximum coverage is a classic submodular optimization task on graphs with many real-world applications such as community detection [GGT14],



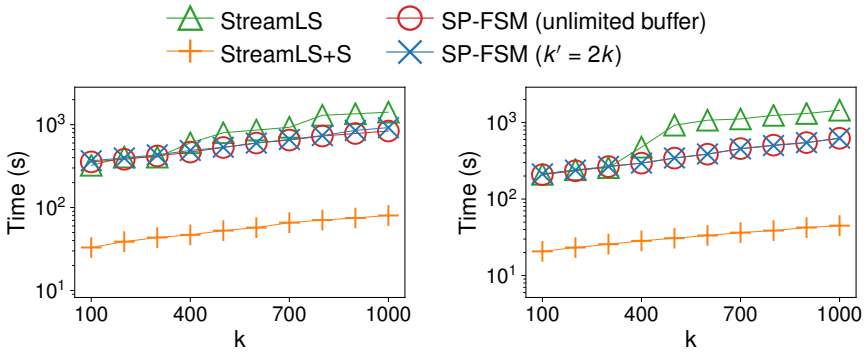
(a) POKEC (Gender) (b) POKEC (Age)

Figure 7.3: Running time of multi-pass algorithms on POKEC. In what follows, we only present the running time for PR because the running time for ER is similar to that for PR.

influence maximization [KKT03], and web monitoring [SG09]. The goal of this task is to select a small subset of nodes that covers a large portion of nodes in a graph. Formally, given a graph $G = (V, E)$ where $n = |V|$ is the number of nodes and $m = |E|$ is the number of edges, the goal is to find a subset $S \subseteq V$ that maximizes the nodes in the neighborhood of S, i.e., $f(S) = |\cup_{v \in S} N(v)|$ where $N(v)$ is the set of nodes connected to v. It is easy to verify that the above function f is nonnegative, monotone, and submodular.

We perform the experiments for maximum coverage on two graph datasets as follows: (1) **POKEC** is a real dataset on SNAP[1]. It is a directed graph with 1,632,803 nodes and 30,622,564 edges representing the follower/followee relationships among users in Pokec. Each node is associated with a user profile with demographic information. The nodes are partitioned into $l = 2$ groups by gender or $l = 7$ groups by age in our experiments. (2) **SYN** is a set of synthetic graphs generated by the Barabási-Albert model [AB02] with equal number of nodes and edges $n = m$. To test the effect of graph size, we generate different graphs with n (as well as m) ranging from 100k to 1m. The nodes are randomly parti-

[1]https://snap.stanford.edu/data/soc-Pokec.html

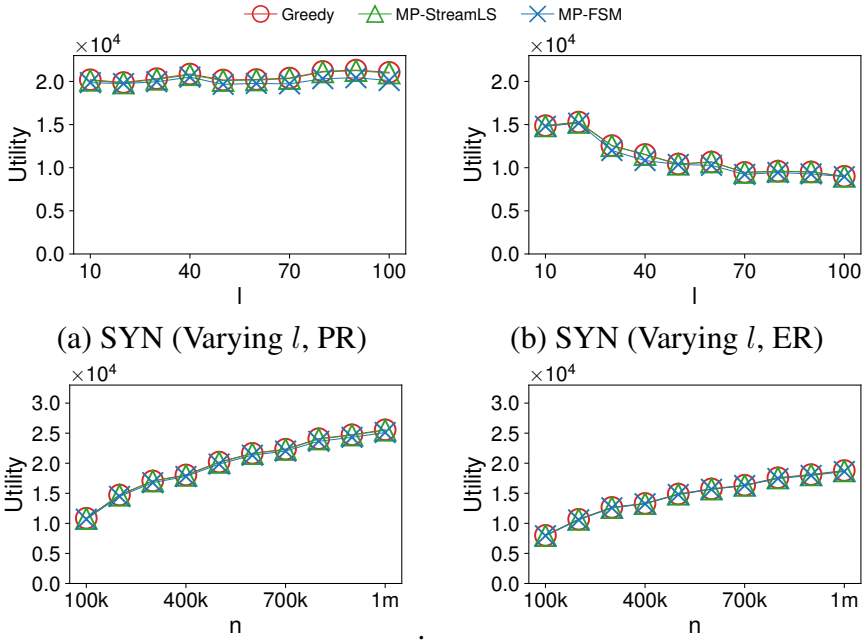


(a) POKEC (Gender) (b) POKEC (Age)

Figure 7.4: Running time of single-pass algorithms on POKEC.

tioned into l groups and the group sizes follow a Zipf's distribution with parameter $s = 2$. By default, we set the number l of groups to 10. To test the effect of l, we fix $n = 500$k and vary l from 10 to 100.

In the first set of experiments, we evaluate the performance of GREEDY, MP-STREAMLS, and MP-FSM in a multi-pass streaming setting. We range the total cardinality constraint $k = \sum_i k_i$ from 100 to $1,000$ and use both *proportional representation* (PR) and *equal representation* (ER) to set k_i for the different groups as fairness constraints. The solution utilities and running time on the POKEC dataset are presented in Figures 7.1 and 7.3, respectively. First of all, "the price of fairness" – i.e., the loss in utility caused by the fairness constraints, is marginal for PR in both cases of Gender and Age groups, and ER in the case of Gender groups where the groups are few and roughly balanced (e.g., 51% female vs. 49% male on POKEC). However, if the groups are highly imbalanced (e.g., 7 age groups on POKEC), enforcing equal representation leads to significant losses in utilities (see Figure **??**). Furthermore, MP-FSM outperforms GREEDY and MP-STREAMLS in running time and solution utility in almost all cases. It runs up to 19 and 567 times faster than GREEDY and MP-STREAMLS, respectively. Meanwhile, its solution utilities are always nearly equal to (at least 99% of) those of GREEDY and consistently better (up to 10% higher) than those of MP-STREAMLS.



(a) SYN (Varying l, PR)

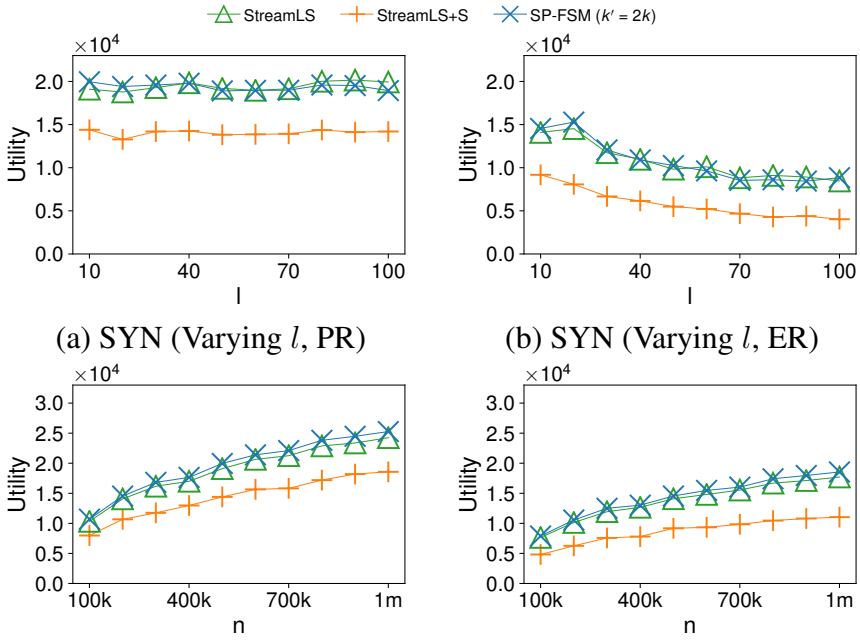
(b) SYN (Varying l, ER)

(c) SYN (Varying n, PR)

(d) SYN (Varying n, ER)

Figure 7.5: Solution utilities of multi-pass algorithms with varying dataset size n and number of groups l.

In the second set of experiments, we evaluate the performance of STREAMLS, STREAMLS+S, and SP-FSM with unlimited and bounded (i.e., $k' = 2k$) buffer sizes in a single-pass streaming setting. We also vary k from 100 to $1,000$ and use both PR and ER as fairness constraints. The experimental results on the POKEC dataset are illustrated in Figures 7.2 and 7.4. Firstly, we observe that the utilities of the solutions provided by both STREAMLS and SP-FSM are typically around 10% lower than those of the solutions of GREEDY. This can be seen as "the price of streaming" – i.e., the loss in utilities for restricting data access only to a single pass over the stream. Secondly, the solution quality of SP-FSM is consistently equivalent to or better than that of STREAMLS. Meanwhile, the efficiency of SP-FSM in terms of running time is also consistently higher than that



(a) SYN (Varying l, PR)

(b) SYN (Varying l, ER)

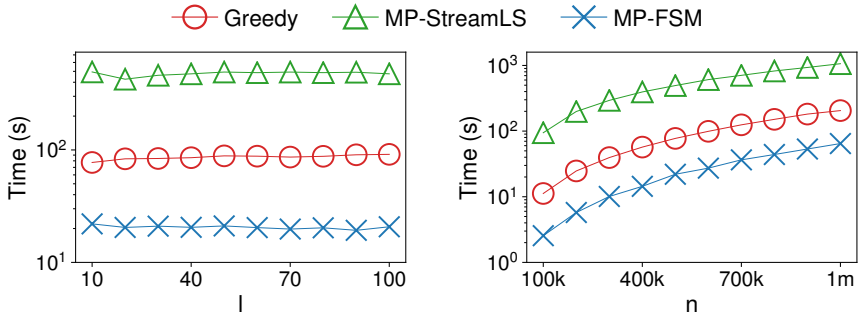
(c) SYN (Varying n, PR)

(d) SYN (Varying n, ER)

Figure 7.6: Solution utilities of single-pass algorithms with varying dataset size n and number of groups l.

of StreamLS, particularly so for larger values of k. Thirdly, when the buffer size is limited to $2k$, the performance of SP-FSM is nearly equivalent to that of SP-FSM with unlimited buffer sizes, which confirms the effectiveness of the buffer management strategy we propose. Fourthly, the subsampling technique of StreamLS+S does not work well in our scenario: although it leads to obvious improvements in efficiency, its solution quality is significantly inferior to any other algorithm.

In the third set of experiments, we test the scalability of different algorithms with varying the number l of groups and the dataset size n on synthetic datasets **SYN**. The solution utilities of multi-pass algorithms are shown in Figure 7.5. We observe that the utilities of different algorithms remain stable for PR but decrease for ER when the number l of groups

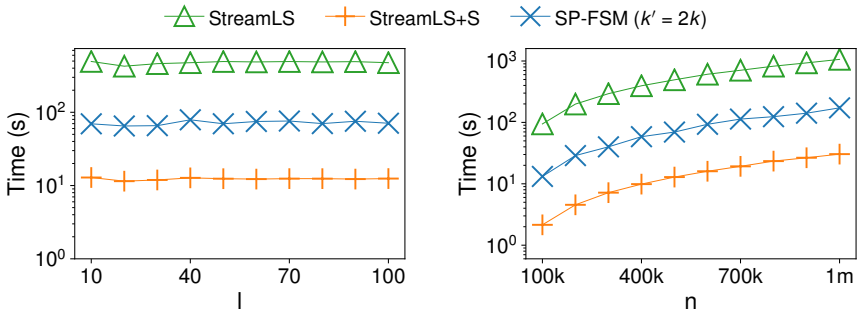


(a) SYN (Varying l) (b) SYN (Varying n)

Figure 7.7: Running time of multi-pass algorithms with varying dataset size n and number of groups l.

increases. This is also an evidence of "the price of fairness", – i.e., enforcing the selection of an equal number of items from highly unbalanced groups (note that the group sizes on SYN follow a Zipf's distribution) causes significant utility losses. Nevertheless, the utilities of different algorithms grow with increasing n as expected. At the same time, the solution utilities of GREEDY, MP-STREAM-LS, and MP-FSM are generally close to each other with varying l and n. The running time of multi-pass algorithms are shown in Figure 7.7. The running time generally keeps steady for different values of l and grows near linearly with increasing n. Meanwhile, MP-FSM runs nearly 10 and 100 times faster than GREEDY and MP-STREAM-LS, respectively, for different values of l and n.

The corresponding results for single-pass algorithms on synthetic datasets are presented in Figures 7.6 and 7.8, respectively. Since SP-FSM with unlimited buffer size shows nearly identical performance to SP-FSM with buffer size $2k$, we omit its results here. In general, we observe the same trends as the multi-pass case with varying l and n. For different values of l and n, the solution quality of SP-FSM and STREAMLS is close to each other, but SP-FSM runs much faster than STREAMLS. With the benefit of subsampling, STREAMLS+S has much higher efficiency than SP-FSM and STREAMLS. Nevertheless, its solution quality is obviously



(a) SYN (Varying l) (b) SYN (Varying n)

Figure 7.8: Running time of single-pass algorithms with varying dataset size n and number of groups l.

worse than them at the same time.

In summary for the application of graph coverage, our results demonstrate that our proposed algorithms MP-FSM and SP-FSM manage to pay a small 'price' for the restrictions of the setting (fairness constraints and streaming data access) and, compared to existing algorithms, they exhibit an excellent combination of performance both in terms of running time and utility.

7.5.2 Personalized Recommendation

The personalized recommendation problem has been used for benchmarking submodular maximization algorithms in [MBN+17, NTM+18]. Its goal is to select a subset S of k items that is both relevant to a given user u and well represents all items in the collection V. Formally, each query user u and each item v in the collection V are denoted by feature vectors in $\mathbb{R}^d$. The relevance between users and items is computed by the inner product of their feature vectors. The objective function is defined as follows:

$$f(S) = \lambda \cdot \sum_{v' \in V} \max_{v \in S} \langle v', v \rangle + (1 - \lambda) \cdot \sum_{v \in S} \langle u, v \rangle$$

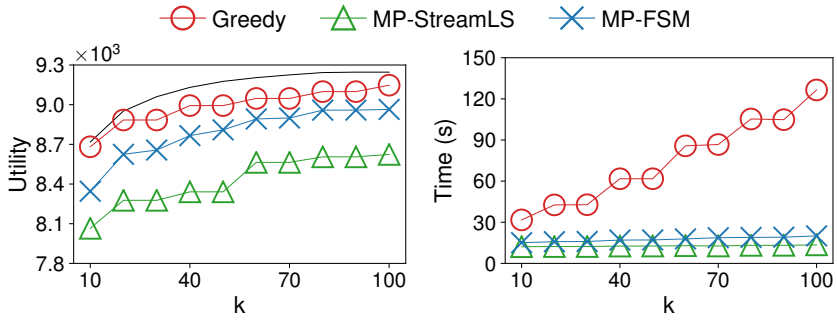


Figure 7.9: Results of multi-pass algorithms on MovieLens. The solution utilities of GREEDY without fairness constraints are plotted as black lines.

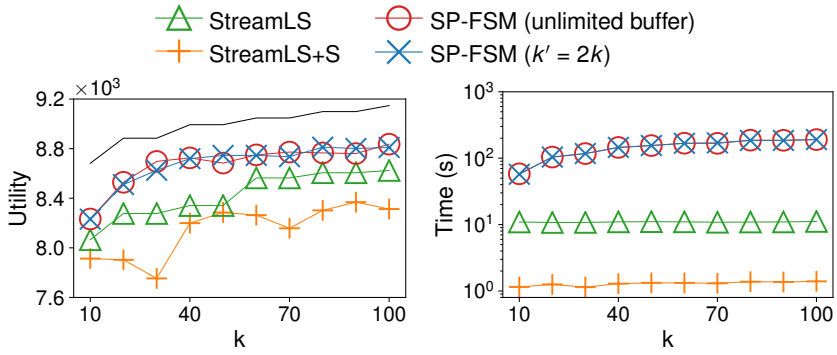


Figure 7.10: Results of single-pass algorithms on MovieLens. The solution utilities of GREEDY are plotted as black lines.

and, again, it is nonnegative, monotone and submodular. The first term is used to measure how well a subset S represents the collection V; the second term denotes the relevance of S to user u; and the parameter λ trades off between both terms. We set $\lambda = 0.75$ following [MBN+17, NTM+18] in our experiments.

We perform the experiments for personalized recommendation on the **MovieLens** dataset[2]. It contains $3,883$ items (movies) and $6,040$ users with 1 million user ratings. We denote each item and user as a 50-

[2]https://grouplens.org/datasets/movielens/

dimensional vector by performing Nonnegative Matrix Factorization (NMF) [WZ13] on the user rating data. The items (movies) are partitioned into $l = 18$ groups according to *genre*.

We evaluate the performance of multi-pass algorithms by ranging k from 10 to 100 in Figure 7.9. Because the results for ER are similar to those for PR, we omit the results for ER here. Since the number l of groups is large compared with k, the utility losses caused by fairness constraints (for both PR and ER) are more significant than those in *maximum coverage*. Among the multi-pass algorithms, GREEDY runs the slowest but achieves the best solution quality. Moreover, MP-FSM has higher efficiency than GREEDY, especially when k becomes larger. Meanwhile, it provides solutions of at least 96% utilities of those of GREEDY. Although MP-STREAMLS runs faster than MP-FSM and GREEDY because of fewer number of solution updates, its solution quality becomes worse as well. We evaluate the performance of single-pass algorithms by ranging k from 10 to 100 in Figure 7.10. Similar to the case of *maximum coverage*, the solution utilities of SP-FSM (with unlimited and bounded buffer sizes) are around 10% lower than those of GREEDY because only one pass over the stream is permitted. Meanwhile, SP-FSM provides solutions of higher quality than STREAMLS at the expense of longer running time. Finally, STREAMLS+S still brings great improvements in efficiency but leads to obvious losses in solution quality.

7.6 Conclusion

We studied the problem of extracting fair and representative items from data streams. In this paper, we formulated the problem as maximizing monotone submodular functions subject to partition matroid constraints. We first designed a $(\frac{1}{2}-\varepsilon)$-approximation multi-pass streaming algorithm called MP-FSM for the problem. Then, we designed a single-pass streaming algorithm called SP-FSM for the problem. SP-FSM had the same $(\frac{1}{2}-\varepsilon)$-approximation ratio as MP-FSM when an unlimited buffer size is permitted, which improved the best-known $\frac{1}{4}$-approximation ratio in the

literature. We further considered the practical implementation of SP-FSM with bounded buffer sizes. Finally, extensive experimental results on two real-world applications confirmed the efficiency, effectiveness, and scalability of our proposed algorithms.

8

Conclusions

This manuscript advances the state of the art in the direction of studying and mitigating harmful algorithmic biases in presence of a recommendation algorithm. In particular, in the first part, we emphasize how network structures can be leveraged for analyzing phenomena due to the interactions between users and algorithmic recommendations.

In the contest of Online Social Networks we investigated, through a series of empirical evidences, the impact of network homophily over the output of a People Recommender System. Our results showed how the more homophilic is a subgroup, the more unfairness would be generated by the recommenders, with a tendency to benefit the homophilic users. Afterwards, this first study led to analyze the same category of algorithms with a perspective on the long-term effects. For this reason, we designed a simulation model able to reproduce the "feedback loop" stimulated by sequential interactions between users and recommendation algorithms. In this case, we evaluated the impact of the homophily not only on the algorithmic suggestions, but also on the original network, displaying the interplay between original topology, user behavior and recommendation algorithms.

Overall, the insights grasped through both the empirical study and the simulation model, allowed us to design methods able to reduce prominent issues such as radicalization in sequential output and unfairness in

an online fashion (e.g. streaming data). Indeed, we first designed an algorithm able to reduce the harmful behavior of "radicalization" on sequential recommendations (e.g. "what-to-watch-next')'. We defined a new metric of recommendations' segregation to propose post-processing strategies of rewiring, which leveraging the network structure, can change the recommendation graph, maintaining a high level of relevance. Then, to further explore bias-aware strategies in an online fashion, we designed techniques able to reduce unfairness of recommendations in the presence of streaming data.

8.1 Limitations & Future Work

Even if our contributions led to four major scientific contributions published in premium venues, they open up to different follow-ups and potential future work.

8.1.1 People Recommender Systems

Bias in PRS. In the 4th Chapter we design a metric of exposure inequalities accounting for the relative size of the two subgroups in the network. Although the wide range of scenarios that have been covered along this work, we point out here several limitations. First, our analysis is purely empirical, leaving space to further investigate analytically how homophily leads to disparate exposure. We expect our work may initiate more contributions in that direction. Also, designing alternative exposure metrics, integrating graph characteristics into the exposure (e.g. in-degree, node centralities) might be a natural follow-up. In addition, this kind of study would clarify different kinds of network inequalities. Nevertheless, through the tools and the analysis we have applied, we do not try to infer the reason behind the observed phenomena, since our findings are mainly driven by empirical evidences. We highlight algorithmic biases expressed in terms of exposure, in a static "single round" of recommendations but saying nothing about the temporal effect of the algorithms. The intro-

duced exposure metric accounts for distribution of recommended users, but does not provide any information regarding the ones receiving the suggestion. In this way, this metric tells nothing about which source a group benefitted from, in terms of accumulated exposure. Synthetic data has been designed to reproduce, user homophily and rich-get-richer effect, and the choice of using the biased preferential attachment is due to its proven capacity to reproduce quite well social network activities [BA99]. Analogously, the experiments are designed assuming a sensitive attribute that allows us to split nodes only in two subgroups. In practice, user demographics may present more than two attribute values (e.g. age, education, etc...) and, in those cases, new definitions of disparate exposure and homophily are needed. In our setting, we assume a power-law distribution leading to the graph generation. Nevertheless, this assumption may sound too tight, and a natural open question to address in the future would be the comparison of results produced by different data generation processes. Also, extending this analysis to other models will open to other use-cases where homophily would raise in other forms, like job platforms interaction networks or research collaboration graphs. In those cases, new definitions of homophily are needed, along with different types of inequality analysis. In particular, the focus would not be on the distribution of exposure but on different kinds of bottleneck for information access, since recommendations may alter segregation in the social graph for different subgroups. After having characterized the inequality in network given by recommendation algorithms, there is an urgent need to design homophily-aware recommendation algorithms, able to deal with the interplay between utility, unfairness and homophily. In particular, there is an urgency of introducing techniques able to reduce unfairness in exposure, but at the same time without loosing in relevance and considering different levels of individual homophily.

Long-Term Analysis. In the 5th Chapter, we present insights in the direction of long-term effects of recommendation systems using simulation models. However, disentangling the consequences of recommendation algorithms, user behavior and network topology remains challenging. Hence, we next discuss some limitations of our study, along with some

natural follow-ups. First, the user behaviour included in our work does not consider homophily as a potential factor. In our future effort we plan to investigate how homophily can impact user choices when accepting or rejecting algorithmic recommendations. A more fine-grain analysis at the mesoscale level of communities or subgraphs might be useful to better understand the phenomena at play. Moreover, a natural extension of our work would be a setting where also the graph evolves dynamically, opening to scenarios where homophily may change over time as well as with the partition of minority-majority. Including the organic network growth as part of the simulation would allow the users to have heterogeneous behaviours, since in real-world scenarios each user may express homophily in different ways of interactions. For this reason, verifying our findings through auditing timely tracked interactions between users and recommendation algorithms is needed. Finally, the results of the present study highlight the urgency to include link recommendation algorithms among the key elements when modelling network dynamics. Bias-aware methods able to mitigate exposure would advance the research in the direction of long-term unfairness.

8.1.2 Bias Mitigation

What-To-Watch-Next. In the 6th Chapter, we present a network-based strategy to reduce radicalization in recommendation pathways. The main limitation of our problem formulation is assuming a binary labelling of nodes, which limits each content to one of the two groups (harmful or neutral), which is not always realistic. A natural extension is to not assume binary labels, but relative scores in $[0, 1]$. This change would require to re-define segregation accordingly. Moreover, assuming the recommendation graph to be static and part as input represents a limitation, since this implies that the recommendations are precomputed and static. Also, in our algorithms we do not include any mechanism replicating the monetization framework applied on video sharing platforms. Additionally, our setting can be extended to a scenario where recommendations are generated dynamically in batches and not statically. Our work is also the

first one proposing post-processing strategies to reduce radicalization. A natural follow-up may be designing in-process algorithms able to reduce segregation through radicalization while training the algorithm.
Fairness in Streaming Data. In the 7th Chapter we introduced algorithms to consider group fairness definitions in pipelines involving streaming data. The main limitation of the proposed algorithms regards the problem formulation, since no definition of individual fairness is considered. Also, this setting considers only the fairness perspective from the side of the provider, but not the one of the consumer.

8.2 Ethical Considerations & Implications

The 4th and 5th Chapters sheds light on some ethical key aspects to consider into the design of social networking products, hinting the need to design Online Social Platforms able to considerate existing biases while introducing new features. Through simulation it is possible to test new recommendation algorithms, e.g., to prevent eventual harmful consequences of deploying new features.

In the 6th chapter, we aim at reducing the exposure to radicalized content generated by W2W recommender systems. Our approach does not include any form of censorship, and instead limits algorithmic-induced over-exposure, which is stimulated by biased organic interactions (e.g., the spread of radicalized content through user-user interactions). Our work contributes to raise awareness on the importance of devising policies aimed at reducing harmful algorithmic side effects. Generally, we do not foresee any immediate and direct harmful impacts from this chapter.

Bibliography

[AAB+19] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Augusto Pizzato. Beyond personalization: Research directions in multistakeholder recommendation. *CoRR*, abs/1905.01986, 2019. 4, 21

[AB02] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 2002. 140

[AEF+20] Naor Alaluf, Alina Ene, Moran Feldman, Huy L. Nguyen, and Andrew Suh. Optimal streaming algorithms for submodular maximization with cardinality constraints. In *ICALP*, pages 6:1–6:19, 2020. 120, 122

[AES+20] Joshua Asplund, Motahhare Eslami, Hari Sundaram, Christian Sandvig, and Karrie Karahalios. Auditing race and gender discrimination in online housing markets. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 24–35, 2020. 30

[AFBS21] Alejandro Ariza, Francesco Fabbri, Ludovico Boratto, and Maria Salamó. From the beatles to billie eilish: Connecting provider representativeness and exposure in session-based recommender systems. In *European Conference on Information Retrieval*, pages 201–208. Springer, 2021. 11, 30, 31

[AG05] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005. 3, 27

[AG17] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *J. Econ. Perspect.*, 31(2):211–236, 2017. 93

[AMA+20] Ashton Anderson, Lucas Maystre, Ian Anderson, Rishabh Mehrotra, and Mounia Lalmas. Algorithmic effects on the diversity of consumption on spotify. In *Proceedings of The Web Conference 2020*, pages 2155–2165, 2020. 1

[AMT13] Zeinab Abbassi, Vahab S. Mirrokni, and Mayur Thakur. Diversity maximization under matroid constraints. In *KDD*, pages 32–40, 2013. 121

[AS19] Victor Amelkin and Ambuj K. Singh. Fighting opinion control in social networks via link recommendation. In *KDD*, pages 677–685, 2019. 95

[BA99] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. 22, 151

[BAFL21] Nathan Bartley, Andres Abeliuk, Emilio Ferrara, and Kristina Lerman. Auditing algorithmic bias on twitter. In *13th ACM Web Science Conference 2021*, pages 65–73, 2021. 27, 28

[BCD+18] Elisabetta Bergamini, Pierluigi Crescenzi, Gianlorenzo D'Angelo, Henning Meyerhenke, Lorenzo Severini, and Yllka Velaj. Improving the betweenness centrality of a node by adding links. *ACM J. Exp. Algorithmics*, 23, 2018. 95

[BFM19] Ludovico Boratto, Gianni Fenu, and Mirko Marras. The effect of algorithmic bias on recommender systems for massive open online courses. In *European Conference on Information Retrieval*, pages 457–472. Springer, 2019. 30

[BGW18] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*, pages 405–414, 2018. 92, 97

[BH19] Avishek Bose and William Hamilton. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*, pages 715–724. PMLR, 2019. 32

[BHN19] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. http://www.fairmlbook.org. 23

[BLM14] Danah Boyd, Karen Levy, and Alice Marwick. The networked nature of algorithmic discrimination. *Data and Discrimination: Collected Essays. Open Technology Institute*, 2014. 22

[BMKK14] Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In *KDD*, pages 671–680, 2014. 119, 120, 122, 124, 126, 127

[BS14] Solon Barocas and Andrew D. Selbst. Big Data's Disparate Impact. *SSRN eLibrary*, 2014. 23

[Bur00] Ronald S Burt. The network structure of social capital. *Research in organizational behavior*, 22:345–423, 2000. 22

[Bur17] Robin Burke. Multisided fairness for recommendation. *CoRR*, abs/1707.00093, 2017. 24

[BY18] Ricardo Baeza-Yates. Bias on the web. *Communications of the ACM*, 61(6):54–61, 2018. 21

[BY20] Ricardo Baeza-Yates. Bias in search and recommender systems. In *Fourteenth ACM Conference on Recommender Systems*, pages 2–2, 2020. 27

[CAT14] Hau Chan, Leman Akoglu, and Hanghang Tong. Make it or break it: Manipulating robustness in large networks. In *SDM*, pages 325–333, 2014. 95

[CDR21] Mihaela Curmei, Sarah Dean, and Benjamin Recht. Quantifying availability and discovery in recommender systems via stochastic reachability. In *ICML*, pages 2265–2275, 2021. 95

[CDSV16] Pierluigi Crescenzi, Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. Greedily improving our own closeness centrality in a network. *ACM Trans. Knowl. Discov. Data*, 11(1):9:1–9:32, 2016. 95

[CFG20] Matteo Castiglioni, Diodato Ferraioli, and Nicola Gatti. Election control in social networks via edge addition or removal. In *AAAI*, pages 1878–1885, 2020. 95

[CGQ15] Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming algorithms for submodular function maximization. In *ICALP*, pages 318–330, 2015. 121, 123, 124, 138

[Cha12] Satya R Chakravarty. *Ethical social index numbers*. Springer Science & Business Media, 2012. 52

[CHJ+17] T.-H. Hubert Chan, Zhiyi Huang, Shaofeng H.-C. Jiang, Ning Kang, and Zhihao Gavin Tang. Online submodular maximization with free disposal: Randomization beats $\frac{1}{4}$ for partition matroids. In *SODA*, pages 1204–1223, 2017. 123

[Cho17] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *CoRR*, abs/1703.00056, 2017. 23

[CK15] Amit Chakrabarti and Sagar Kale. Submodular maximization meets streaming: matchings, matroids, and more. *Math. Program.*, 154(1-2):225–247, 2015. 121, 123, 138

[CKR20] Ashish Chiplunkar, Sagar Kale, and Sivaramakrishnan Natarajan Ramamoorthy. How to solve fair k-center in massive data models. In *ICML*, pages 6887–6896, 2020. 123

[CKS+18] L. Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth K. Vishnoi. Fair and diverse dpp-based data summarization. In *ICML*, pages 715–724, 2018. 120, 123

[CKSV19] L Elisa Celis, Sayash Kapoor, Farnood Salehi, and Nisheeth Vishnoi. Controlling polarization in personalization: An algorithmic framework. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 160–169, 2019. 29

[CLB18] Xi Chen, Jefrey Lijffijt, and Tijl De Bie. Quantifying and minimizing risk of conflict in social networks. In *KDD*, pages 1197–1205, 2018. 94

[CM20a] Uthsav Chitra and Christopher Musco. Analyzing the impact of filter bubbles on social network polarization. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 115–123, 2020. 28

[CM20b] Uthsav Chitra and Christopher Musco. Analyzing the impact of filter bubbles on social network polarization. In *WSDM*, pages 115–123, 2020. 94

[CMMB21] Federico Cinus, Marco Minici, Corrado Monti, and Francesco Bonchi. The effect of people recommenders on echo chambers and polarization. 2021. 69, 95

[CMV20] L Elisa Celis, Anay Mehrotra, and Nisheeth K Vishnoi. Interventions for ranking in the presence of implicit bias. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 369–380, 2020. 31

[CNPS11] Gabriele Capannini, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. Efficient diversification of web search results. *arXiv preprint arXiv:1105.4255*, 2011. 33

[CR18] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *CoRR*, abs/1810.08810, 2018. 23

[CR20] Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Commun. ACM*, 63(5):82–89, 2020. 120

[CRF+11] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro

Flammini. Political polarization on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, pages 89–96, 2011. 27

[CSE18] Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 224–232, 2018. 28

[CZTR08] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, page 87–94, New York, NY, USA, 2008. Association for Computing Machinery. 76

[Das18] Jeffrey Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. In *Ethics of Data and Analytics*, pages 296–299. Auerbach Publications, 2018. 2

[DGM10] Elizabeth M. Daly, Werner Geyer, and David R. Millen. The network effects of recommending social connections. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, pages 301–304, 2010. 38, 41

[DHP+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 214–226, 2012. 24, 56

[DMP+16] Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, Karin Pasini, and Roberto Pasolini. Job recommendation from semantic similarity of linkedin users' skills. In *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*, pages 270–277. SciTePress, 2016. 37

[DOS19] Gianlorenzo D'Angelo, Martin Olsen, and Lorenzo Severini. Coverage centrality maximization in undirected networks. In *AAAI*, pages 501–508, 2019. 95

[DRR20] Sarah Dean, Sarah Rich, and Benjamin Recht. Recommendations and user agency: the reachability of collaboratively-filtered information. In *FAT**, pages 436–445, 2020. 95

[DSB+19] Abhisek Dash, Anurag Shandilya, Arindam Biswas, Kripabandhu Ghosh, Saptarshi Ghosh, and Abhijnan Chakraborty. Summarizing user-generated textual content: Motivation and methods for fairness in algorithmic summaries. *Proc. ACM Hum. Comput. Interact.*, 3(CSCW):172:1–172:28, 2019. 120, 123

[EBD19] Michael D. Ekstrand, Robin Burke, and Fernando Diaz. Fairness and discrimination in retrieval and recommendation. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 1403–1404, New York, NY, USA, 2019. ACM. 21, 24, 40

[EK10] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge university press, 2010. 22

[ELVZ17] Alessandro Epasto, Silvio Lattanzi, Sergei Vassilvitskii, and Morteza Zadimoghaddam. Submodular optimization

over sliding windows. In *WWW*, pages 421–430, 2017. 119, 122, 124

[ESM19] Farzad Eskandanian, Nasim Sonboli, and Bamshad Mobasher. Power of the few: Analyzing the impact of influential users in collaborative recommender systems. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, pages 225–233, 2019. 31

[ETA+18] Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81, pages 172–186. PMLR, 23–24 Feb 2018. 30, 31

[FBBC20] Francesco Fabbri, Francesco Bonchi, Ludovico Boratto, and Carlos Castillo. The effect of homophily on disparate visibility of minorities in people recommender systems. In *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020*, pages 165–175. AAAI Press, 2020. 8, 65, 67, 68, 70, 72, 73, 74, 95

[FCBC21] Francesco Fabbri, Maria Luisa Croci, Francesco Bonchi, and Carlos Castillo. Exposure inequality in people recommender systems: The long-term effects, 2021. 9, 95

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. 120, 124

[FKK18] Moran Feldman, Amin Karbasi, and Ehsan Kazemi. Do less, get more: Streaming submodular maximization with

subsampling. In *NeurIPS*, pages 730–740, 2018. 121, 123, 124, 138

[FNW78] Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. An analysis of approximations for maximizing submodular set functions—II. In Michel L. Balinski and Alan J. Hoffman, editors, *Polyhedral Combinatorics*, pages 73–87. Springer Berlin Heidelberg, 1978. 121, 122, 125, 138

[FVD+16] Emilio Ferrara, Onur Varol, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Commun. ACM*, 59(7):96–104, 2016. 93

[FWB+22] Francesco Fabbri, Yanhao Wang, Francesco Bonchi, Carlos Castillo, and Michael Mathioudakis. Rewiring what-to-watch-next recommendations to reduce radicalization pathways. In *Proceedings of the ACM Web Conference 2022*, pages 2719–2728, 2022. 10

[GAC+13] Ido Guy, Uri Avraham, David Carmel, Sigalit Ur, Michal Jacovi, and Inbal Ronen. Mining expertise and interests from social media. In *22nd International World Wide Web Conference*, pages 515–526. International World Wide Web Conferences Steering Committee / ACM, 2013. 37

[GGLM19] Fabrizio Germano, Vicenç Gómez, and Gaël Le Mens. The few-get-richer: a surprising consequence of popularity-based rankings? In *The World Wide Web Conference*, pages 2764–2770, 2019. 85

[GGM19] Fabrizio Germano, Vicenç Gómez, and Gaël Le Mens. The few-get-richer: a surprising consequence of popularity-based rankings? In *The World Wide Web Conference, WWW 2019*, pages 2764–2770. ACM, 2019. 43

[GGT14] Esther Galbrun, Aristides Gionis, and Nikolaj Tatti. Overlapping community detection in labeled graphs. *Data Min. Knowl. Discov.*, 28(5-6):1586–1610, 2014. 139

[GJ79] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. 98

[GJCK13] Pedro Henrique Calais Guerra, Wagner Meira Jr., Claire Cardie, and Robert Kleinberg. A measure of polarization on social media networks based on community boundaries. In *ICWSM*, pages 215–224, 2013. 94

[GK10] Ryan Gomes and Andreas Krause. Budgeted nonparametric learning from data streams. In *ICML*, page 391–398, 2010. 119, 120, 122, 124

[GLG+21] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, and et al. Towards long-term fairness in recommendation. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, March 2021. 69

[GMGM16] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy in social media. In *WSDM*, pages 33–42, 2016. 94

[GMGM17] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Reducing controversy by connecting opposing views. In *WSDM*, pages 81–90, 2017. 94

[GMGM18] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying

controversy on social media. *ACM Transactions on Social Computing*, 1(1):1–27, 2018. 74

[GMS13] David García, Pavlin Mavrodiev, and Frank Schweitzer. Social resilience in online communities: the autopsy of friendster. In *Conference on Online Social Networks, COSN'13, Boston, MA, USA, October 7-8, 2013*, pages 39–50, 2013. 40

[GP16] Ido Guy and Luiz Pizzato. People recommendation tutorial. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 431–432. ACM, 2016. 17, 37

[GRW09] Ido Guy, Inbal Ronen, and Eric Wilcox. Do you know?: recommending people to invite into your social network. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 77–86. ACM, 2009. 37

[GSB21] David García-Soriano and Francesco Bonchi. Maxmin-fair ranking: individual fairness under group-fairness constraints. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 436–446, 2021. 31

[GW17] Venkata Rama Kiran Garimella and Ingmar Weber. A long-term analysis of polarization on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 528–531, 2017. 22

[HKV08] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008. 20, 45, 109

[HKW+14] Bradford Heap, Alfred Krzywicki, Wayne Wobcke, Mike Bain, and Paul Compton. Combining career progression

and profile matching in a job recommender system. In *Trends in Artificial Intelligence - 13th Pacific Rim International Conference on Artificial Intelligence*, pages 396–408. Springer, 2014. 37

[HL10] Willem Halffman and Loet Leydesdorff. Is inequality among universities increasing? gini coefficients and the elusive rise of elite universities. *Minerva*, 48(1):55–72, 2010. 82

[HLT16] Kai-Hsiang Hsu, Cheng-Te Li, and Chien-Lin Tseng. Who will respond to your requests for instant troubleshooting? In *Proceedings of the Tenth International Conference on Web and Social Media*, pages 591–594. AAAI Press, 2016. 37

[HMRU21] Shahrzad Haddadan, Cristina Menghini, Matteo Riondato, and Eli Upfal. RePBubLik: Reducing polarized bubble radius with link insertions. In *WSDM*, pages 139–147, 2021. 94, 95, 111

[HTBL18] Jevan A Hutson, Jessie G Taft, Solon Barocas, and Karen Levy. Debiasing desire: Addressing bias & discrimination on intimate platforms. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–18, 2018. 30

[HTW20] Chien-Chung Huang, Theophile Thiery, and Justin Ward. Improved multi-pass streaming algorithms for submodular maximization with matroid constraints. In *APPROX/RANDOM*, pages 62:1–62:19, 2020. 123, 138

[HVR+16] Viet Ha-Thuc, Ganesh Venkataraman, Mario Rodriguez, Shakti Sinha, Senthil Sundaram, and Lin Guo. Personalized expertise search at linkedin. *CoRR*, abs/1602.04572, 2016. 37

[IMR21] Ruben Interian, Jorge R. Moreno, and Celso C. Ribeiro. Polarization reduction by minimum-cardinality edge additions: Complexity and integer programming approaches. *Int. Trans. Oper. Res.*, 28(3):1242–1264, 2021. 94

[JCL+19] Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 383–390, 2019. 69

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002. 92, 97

[JLNN20] Matthew Jones, Huy Lê Nguyên, and Thy Nguyen. Fair k-centers via maximum matching. In *ICML*, pages 7460–7469, 2020. 121, 123

[Joa02] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002. 76

[KAAS18] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Recommendation independence. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, pages 187–201, 2018. 24, 32

[KAM19] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *ICML*, pages 3448–3457, 2019. 121, 123

[KDS14] Elias Boutros Khalil, Bistra Dilkina, and Le Song. Scalable diffusion-aware optimization of network topology. In *KDD*, pages 1226–1235, 2014. 94

[KG14] Andreas Krause and Daniel Golovin. Submodular function maximization. In Lucas Bordeaux, Youssef Hamadi, and Pushmeet Kohli, editors, *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge University Press, 2014. 122

[KGW+18] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. Homophily influences ranking of minorities in social networks. *Scientific reports*, 8(1):1–12, 2018. 42, 73

[KJJ18] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. News recommender systems–survey and roads ahead. *Information Processing & Management*, 54(6):1203–1227, 2018. 2

[KKT03] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003. 119, 122, 140

[Kle18] Jon Kleinberg. Inherent trade-offs in algorithmic fairness. In *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems*, pages 40–40, 2018. 3

[KMM15] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *CHI*, pages 3819–3828, 2015. 120

[KMR17a] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of

risk scores. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 43:1–43:23, 2017. 3

[KMR17b] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 43:1–43:23, 2017. 23

[KMVV15] Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in MapReduce and streaming. *ACM Trans. Parallel Comput.*, 2(3):14:1–14:22, 2015. 122, 126, 127

[KMZ+19] Ehsan Kazemi, Marko Mitrovic, Morteza Zadimoghaddam, Silvio Lattanzi, and Amin Karbasi. Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity. In *ICML*, pages 3311–3320, 2019. 120, 122, 124, 126, 127

[KSM08] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Minimizing the spread of contamination by blocking links in a network. In *AAAI*, pages 1175–1180, 2008. 94

[KTS+13] Chris J. Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav V. Marathe, and S. S. Ravi. Blocking simple and complex contagion by edge removal. In *ICDM*, pages 399–408, 2013. 94

[KZK18] Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *ICML*, pages 2549–2558, 2018. 123

[LB18] Weiwen Liu and Robin Burke. Personalizing fairness-aware re-ranking. *CoRR*, abs/1809.02921, 2018. 33

[LET15] Long T. Le, Tina Eliassi-Rad, and Hanghang Tong. MET: A fast algorithm for minimizing propagation in large graphs with small eigen-gaps. In *SDM*, pages 694–702, 2015. 94

[Lew18] Rebecca Lewis. Alternative influence: Broadcasting the reactionary right on youtube. Technical report, Data & Society Research Institute, Sep. 2018. 91

[LFS17a] Zhepeng Li, Xiao Fang, and Olivia R Liu Sheng. A survey of link recommendation for social networks: methods, theoretical foundations, and future research directions. *ACM Transactions on Management Information Systems (TMIS)*, 9(1):1–26, 2017. 74

[LFS17b] Zhepeng (Lionel) Li, Xiao Fang, and Olivia R. Liu Sheng. A survey of link recommendation for social networks: Methods, theoretical foundations, and future research directions. *ACM Trans. Manage. Inf. Syst.*, 9(1):1:1–1:26, October 2017. 17, 45

[LKG+07] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne M. Van Briesen, and Natalie S. Glance. Cost-effective outbreak detection in networks. In *KDD*, pages 420–429, 2007. 122, 124

[LKJ+17] Eun Lee, Fariba Karimi, Hang-Hyun Jo, Markus Strohmaier, and Claudia Wagner. Homophily explains perception biases in social networks. *CoRR*, abs/1710.08601, 2017. 42, 57

[LKW+19] Eun Lee, Fariba Karimi, Claudia Wagner, Hang-Hyun Jo, Markus Strohmaier, and Mirta Galesic. Homophily and minority-group size explain perception biases in social networks. *Nature human behaviour*, 3(10):1078–1087, 2019. 68

[LMF+07] Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie Glance, and Matthew Hurst. Patterns of cascading behavior in large blog graphs. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 551–556. SIAM, 2007. 27

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized bert pretraining approach, 2019. 110

[LOR+16a] Rui Liu, Yuanxin Ouyang, Wenge Rong, Xin Song, Cui Tang, and Zhang Xiong. Rating prediction based job recommendation service for college students. In *Computational Science and Its Applications - 16th International Conference, Proceedings*, pages 453–467. Springer, 2016. 37

[LOR+16b] Rui Liu, Yuanxin Ouyang, Wenge Rong, Xin Song, Weizhu Xie, and Zhang Xiong. Employer oriented recruitment recommender service for university students. In *Intelligent Computing Methodologies - 12th International Conference, Proceedings*, pages 811–823. Springer, 2016. 37

[LVKK16] David Laniado, Yana Volkovich, Karolin Kappler, and Andreas Kaltenbrunner. Gender homophily in online dyadic and triadic relationships. *EPJ Data Sci.*, 5(1):19, 2016. 46, 71

[LWD16] Erik M. Lindgren, Shanshan Wu, and Alexandros G. Dimakis. Leveraging sparsity for efficient submodular data summarization. In *NIPS*, pages 3414–3422, 2016. 122, 124

[LY15] Rong-Hua Li and Jeffrey Xu Yu. Triangle minimization in large networks. *Knowl. Inf. Syst.*, 45(3):617–643, 2015. 95

[LZ20] Mark Ledwich and Anna Zaitsev. Algorithmic extremism: Examining youtube's rabbit hole of radicalization. *First Monday*, 25(3), 2020. 3, 91

[MAD+17] Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach, and Emine Yilmaz. Auditing search engines for differential satisfaction across demographics. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 626–633, 2017. 29

[MAP+20] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems, 2020. 65, 69

[MBN+17] Slobodan Mitrovic, Ilija Bogunovic, Ashkan Norouzi-Fard, Jakub Tarnawski, and Volkan Cevher. Streaming robust submodular maximization: A partitioned thresholding approach. In *NIPS*, pages 4557–4566, 2017. 123, 145, 146

[MKK17] Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Deletion-robust submodular maximization: Data summarization with "the right to be forgotten". In *ICML*, pages 2449–2458, 2017. 123

[MM08] Clark McCauley and Sophia Moskalenko. Mechanisms of political radicalization: Pathways toward terrorism. *Terror. Political Violence*, 20(3):415–433, 2008. 91

[MMB+18] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 2243–2251, 2018. 33

[MMG15] Charalampos Mavroforakis, Michael Mathioudakis, and Aristides Gionis. Absorbing random-walk centrality: Theory and algorithms. In *ICDM*, pages 901–906, 2015. 93, 101, 102, 107

[MMS+21] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021. 21

[MMT18] Cameron Musco, Christopher Musco, and Charalampos E. Tsourakakis. Minimizing polarization and disagreement in social networks. In *WWW*, pages 369–378, 2018. 94

[MSB17] Mainack Mondal, Leandro Araújo Silva, and Fabrício Benevenuto. A measurement study of hate speech in social media. In *HT*, pages 85–94, 2017. 94

[MSS+18] Sourav Medya, Arlei Silva, Ambuj K. Singh, Prithwish Basu, and Ananthram Swami. Group centrality maximization via network design. In *SDM*, pages 126–134, 2018. 95

[MSWVW20] Michael A Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. Co-designing checklists to understand organizational challenges and

opportunities around fairness in ai. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020. 2

[MZ17] Shervin Malmasi and Marcos Zampieri. Detecting hate speech in social media. In *RANLP*, pages 467–472, 2017. 94

[Nar18] Arvind Narayanan. Translation tutorial: 21 fairness definitions and their politics. *FAT* 2018*, 2018. 23

[NGL+16] Shirin Nilizadeh, Anne Groggel, Peter Lista, Srijita Das, Yong-Yeol Ahn, Apu Kapadia, and Fabio Rojas. Twitter's glass ceiling: The effect of perceived gender on online visibility. In *Tenth International AAAI Conference on Web and Social Media*, 2016. 38, 42

[NHA19] Jeppe Nørregaard, Benjamin D. Horne, and Sibel Adali. NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *ICWSM*, pages 630–638, 2019. 109

[NHH+14] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, pages 677–686, 2014. 69

[NTM+18] Ashkan Norouzi-Fard, Jakub Tarnawski, Slobodan Mitrovic, Amir Zandieh, Aidasadat Mousavifar, and Ola Svensson. Beyond 1/2-approximation for submodular maximization on massive data streams. In *ICML*, pages 3826–3835, 2018. 119, 120, 122, 124, 126, 145, 146

[NWF78] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for max-

imizing submodular set functions—I. *Math. Program.*, 14(1):265–294, 1978. 119, 122, 124

[Pap15] Manos Papagelis. Refining social graph connectivity via shortcut edge addition. *ACM Trans. Knowl. Discov. Data*, 10(2):12:1–12:35, 2015. 95

[Par11] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. penguin UK, 2011. 22

[PB07] Juyong Park and Albert-László Barabási. Distribution of node characteristics in complex networks. *Proceedings of the National Academy of Sciences*, 104(46):17916–17920, 2007. 44

[PBG11] Manos Papagelis, Francesco Bonchi, and Aristides Gionis. Suggesting ghost edges for a smaller world. In *CIKM*, pages 2305–2308, 2011. 95

[PKS20] Evaggelia Pitoura, Georgia Koutrika, and Kostas Stefanidis. Fairness in rankings and recommenders. In *EDBT*, pages 651–654, 2020. 95

[PPM+22] Gourab K Patro, Lorenzo Porcaro, Laura Mitchell, Qiuyue Zhang, Meike Zehlike, and Nikhil Garg. Fair ranking: a critical review, challenges, and future directions. *arXiv preprint arXiv:2201.12662*, 2022. 31

[PPT15] Nikos Parotsidis, Evaggelia Pitoura, and Panayiotis Tsaparas. Selecting shortcuts for a smaller world. In *SDM*, pages 28–36, 2015. 95

[PPT16] Nikos Parotsidis, Evaggelia Pitoura, and Panayiotis Tsaparas. Centrality-aware link recommendations. In *WSDM*, pages 503–512, 2016. 95

[PPZ+20] Kostantinos Papadamou, Antonis Papasavva, Savvas Zannettou, Jeremy Blackburn, Nicolas Kourtellis, Ilias Leontiadis, Gianluca Stringhini, and Michael Sirivianos. Disturbed YouTube for kids: Characterizing and detecting inappropriate videos targeting young children. In *ICWSM*, pages 522–533, 2020. 94

[PTVF07] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007. 104

[PZB+22] Kostantinos Papadamou, Savvas Zannettou, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Michael Sirivianos. "it is just a flu": Assessing the effect of watch history on youtube's pseudoscientific video recommendations. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 723–734, 2022. 27, 28

[QCJ18] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. Sequence-aware recommender systems. *ACM Comput. Surv.*, 51(4):66:1–66:36, 2018. 1

[RB20] Guilherme Ramos and Ludovico Boratto. Reputation (in) dependence in ranking systems: Demographics influence over output disparities. In *Proceedings of the 43rd international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 2061–2064, 2020. 32

[RBKL20] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 469–481, 2020. 30

[RGC19] Bashir Rastegarpanah, Krishna P. Gummadi, and Mark Crovella. Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 231–239, 2019. 29

[Roo19] Kevin Roose. The making of a YouTube radical. *The New York Times*, 2019. 91

[ROW+20] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio A. F. Almeida, and Wagner Meira Jr. Auditing radicalization pathways on YouTube. In *FAT**, pages 131–141, 2020. 92, 94, 109

[RRS11] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011. 1

[SAPV15] Sudip Saha, Abhijin Adiga, B. Aditya Prakash, and Anil Kumar S. Vullikanti. Approximation algorithms for reducing the spectral radius to control epidemic spread. In *SDM*, pages 568–576, 2015. 94

[SC18] Javier Sanz-Cruzado and Pablo Castells. Enhancing structural diversity in social networks by recommending weak ties. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 233–241, 2018. 20, 45

[SCC18] Javier Sanz-Cruzado and Pablo Castells. Contact recommendations in social networks. In *Collaborative Recommendations: Algorithms, Practical Challenges and Ap-*

plications, pages 519–569. World Scientific Publishing, November 2018. 37

[SD13] Alexandre Spaeth and Michel C. Desmarais. Combining collaborative filtering and text similarity for expert profile recommendations in social websites. In *User Modeling, Adaptation, and Personalization - 21th International Conference*, pages 178–189. Springer, 2013. 37

[SFC+21] David Solans, Francesco Fabbri, Caterina Calsamiglia, Carlos Castillo, and Francesco Bonchi. Comparing equity and effectiveness of different algorithms in an application for the room rental market. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 978–988, 2021. 12, 30

[SG09] Barna Saha and Lise Getoor. On maximum coverage in the streaming model & application to multi-topic blogwatch. In *SDM*, pages 697–708, 2009. 119, 140

[SG21] Harini Suresh and John Guttag. A framework for understanding sources of harm throughout the machine learning life cycle. In *Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–9. 2021. 2

[SHC20] Ana-Andreea Stoica, Jessy Xinyi Han, and Augustin Chaintreau. Seeding network influence in biased networks and the benefits of diversity. In *WWW*, pages 2089–2098, 2020. 119

[SJ18] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 2219–2228, New York, NY, USA, 2018. ACM. 25, 31, 44

[SJC+20] Panagiotis Symeonidis, Andrea Janes, Dmitry Chaltsev, Philip Giuliani, Daniel Morandini, Andreas Unterhuber, Ludovik Coba, and Markus Zanker. Recommending the video to watch next: an offline and online evaluation at youtv. de. In *Fourteenth ACM conference on recommender systems*, pages 299–308, 2020. 17

[SKNS19] Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. Debiasing the human-recommender system feedback loop in collaborative filtering. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 645–651, 2019. 70

[SMDE20] Javier Sánchez-Monedero, Lina Dencik, and Lilian Edwards. What does it mean to'solve'the problem of discrimination in hiring? social, technical and legal perspectives from the uk on automated hiring systems. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 458–468, 2020. 30

[SPGK19] Alina Sîrbu, Dino Pedreschi, Fosca Giannotti, and János Kertész. Algorithmic bias amplifies opinion fragmentation and polarization: A bounded confidence model. *PloS one*, 14(3):e0213246, 2019. 3, 28

[SQM+17] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. Fairness in package-to-group recommendations. In *WWW*, pages 371–379, 2017. 119, 122

[SRC18] Ana-Andreea Stoica, Christopher Riederer, and Augustin Chaintreau. Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 923–932, Republic and Canton

of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee. 38, 42, 67, 73, 74

[SSG16] Jessica Su, Aneesh Sharma, and Sharad Goel. The effect of recommendations on network structure. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1157–1167, 2016. 37, 41, 45, 81

[SSL+17] Xiaoyu Shi, Ming-Sheng Shang, Xin Luo, Abbas Khushnood, and Jian Li. Long-term effects of user preference-oriented recommendation method on the evolution of online system. *Physica A: Statistical Mechanics and its Applications*, 467:490–498, 2017. 69

[SSW+17] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explor.*, 19(1):22–36, 2017. 93

[Ste18] Harald Steck. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems*, pages 154–162, 2018. 32

[TPE+12] Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, and Christos Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *CIKM*, pages 245–254, 2012. 94

[TPS+21] Matus Tomlein, Branislav Pecher, Jakub Simko, Ivan Srba, Robert Moro, Elena Stefancova, Michal Kompan, Andrea Hrckova, Juraj Podrouzek, and Maria Bielikova. An audit of misinformation filter bubbles on youtube: Bubble bursting and recent behavior changes. In *Fifteenth ACM Conference on Recommender Systems*, pages 1–11, 2021. 27, 28

[TPT+20] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. Fairness-aware link analysis. *arXiv preprint arXiv:2005.14431*, 2020. 69

[TRG21] Antonela Tommasel, Juan Manuel Rodriguez, and Daniela Godoy. I want to break free! recommending friends from outside the echo chamber. In *Fifteenth ACM Conference on Recommender Systems*, pages 23–33, 2021. 29

[Vit85] Jeffrey Scott Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985. 128, 132

[VMG16] Julita Vassileva, Gordon I. McCalla, and Jim E. Greer. From small seeds grow fruitful trees: How the phelps peer help system stimulated a diverse and innovative research agenda over 15 years. *I. J. Artificial Intelligence in Education*, 26(1):431–447, 2016. 37

[VSFC21] Suresh Venkatasubramanian, Carlos Scheidegger, Sorelle Friedler, and Aaron Clauset. Fairness in networks: Social capital, information access, and interventions. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4078–4079, 2021. 22

[WFM21] Yanhao Wang, Francesco Fabbri, and Michael Mathioudakis. Fair and representative subset selection from data streams. In *Proceedings of the Web Conference 2021*, pages 1340–1350, 2021. 10

[WGJ+21] Christo Wilson, Avijit Ghosh, Shan Jiang, Alan Mislove, Lewis Baker, Janelle Szary, Kelly Trindel, and Frida Polli. Building and auditing fair algorithms: A case study in candidate screening. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 666–677, 2021. 30

[WLT19] Yanhao Wang, Yuchen Li, and Kian-Lee Tan. Efficient representative subset selection over sliding windows. *IEEE Trans. Knowl. Data Eng.*, 31(7):1327–1340, 2019. 122

[WSK+17] Claudia Wagner, Philipp Singer, Fariba Karimi, Jürgen Pfeffer, and Markus Strohmaier. Sampling from social networks with attributes. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1181–1190. ACM, 2017. 47

[WW18] Bari Weiss and Damon Winter. Meet the renegades of the intellectual dark web. *The New York Times*, 2018. 91

[WWB+21] Yuyan Wang, Xuezhi Wang, Alex Beutel, Flavien Prost, Jilin Chen, and Ed H Chi. Understanding and improving fairness-accuracy trade-offs in multi-task learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1748–1757, 2021. 3

[WWRM20] Tomasz Was, Marcin Waniek, Talal Rahwan, and Tomasz P. Michalak. The manipulability of centrality measures - an axiomatic approach. In *AAMAS*, pages 1467–1475, 2020. 95

[WWY15] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1235–1244, 2015. 2

[WZ13] Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Trans. Knowl. Data Eng.*, 25(6):1336–1353, 2013. 147

[YH17] Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. *Advances in neural information processing systems*, 30, 2017. 32

[YHT+21] Sirui Yao, Yoni Halpern, Nithum Thain, Xuezhi Wang, Kang Lee, Flavien Prost, Ed H. Chi, Jilin Chen, and Alex Beutel. Measuring Recommender System Effects with Simulated Users. *arXiv e-prints*, page arXiv:2101.04526, January 2021. 65, 69, 75

[YLW+19] Ruidong Yan, Yi Li, Weili Wu, Deying Li, and Yongcai Wang. Rumor blocking through online link deletion on social networks. *ACM Trans. Knowl. Discov. Data*, 13(2):16:1–16:26, 2019. 94

[ZC20] Meike Zehlike and Carlos Castillo. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020*, pages 2849–2855, 2020. 31

[ZCWL18] Weijie Zhu, Chen Chen, Xiaoyang Wang, and Xuemin Lin. K-core minimization: An edge manipulation approach. In *CIKM*, pages 1667–1670, 2018. 95

[ZHC18] Ziwei Zhu, Xia Hu, and James Caverlee. Fairness-aware tensor-based recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1153–1162, 2018. 32

[ZHW+19] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. Recommending what video to watch next: a multitask ranking system. In *RecSys*, pages 43–51, 2019. 1, 17, 91

[ZML+16] Mingyu Zhang, Jian Ma, Zhiying Liu, Jianshan Sun, and Thushari Silva. A research analytics framework-supported recommendation approach for supervisor selection. *BJET*, 47(2):403–420, 2016. 37

[ZSW+19] Junzhou Zhao, Shuo Shang, Pinghui Wang, John C. S. Lui, and Xiangliang Zhang. Submodular optimization over streams with inhomogeneous decays. In *AAAI*, pages 5861–5868, 2019. 122

[ZYS21] Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000*, 2021. 31