

GNNUERS: Fairness Explanation in GNNs for Recommendation via Counterfactual Reasoning

Giacomo Medda
University of Cagliari
Cagliari, Italy
giacomo.medda@unica.it

Francesco Fabbri
Spotify
Barcelona, Spain
francescof@spotify.com

Mirko Marras
University of Cagliari
Cagliari, Italy
mirko.marras@acm.org

Ludovico Boratto
University of Cagliari
Cagliari, Italy
ludovico.boratto@acm.org

Mihnea Tufis
Eurecat
Barcelona, Spain
mihnea.tufis@eurecat.org

Gianni Fenu
University of Cagliari
Cagliari, Italy
fenu@unica.it

ABSTRACT

In recent years, personalization research has been delving into issues of explainability and fairness. While some techniques have emerged to provide post-hoc and self-explanatory individual recommendations, there is still a lack of methods aimed at uncovering unfairness in recommendation systems beyond identifying biased user and item features. This paper proposes a new algorithm, GNNUERS, which uses counterfactuals to pinpoint user unfairness explanations in terms of user-item interactions within a bi-partite graph. By perturbing the graph topology, GNNUERS reduces differences in utility between protected and unprotected demographic groups. The paper evaluates the approach using four real-world graphs from different domains and demonstrates its ability to systematically explain user unfairness in three state-of-the-art GNN-based recommendation models. This perturbed network analysis reveals insightful patterns that confirm the nature of the unfairness underlying the explanations. The source code and preprocessed datasets are available at bit.ly/GNNUERS.

KEYWORDS

Recommender Systems, User Fairness, Explanation, Graph Neural Networks, Counterfactual Reasoning

ACM Reference Format:

Giacomo Medda, Francesco Fabbri, Mirko Marras, Ludovico Boratto, Mihnea Tufis, and Gianni Fenu. 2023. GNNUERS: Fairness Explanation in GNNs for Recommendation via Counterfactual Reasoning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

As recommender systems become more and more effective and sophisticated, the complexity of their functioning increases dramatically. The recommendations of novel systems improve the satisfaction of the users, but their lack of interpretability lays the groundwork for worrying questions [17].

The issue of interpretability comes in addition with the prominent importance of preserving properties that go beyond recommendation effectiveness, such as trustworthiness [39], fairness [41],

and explainability [49]. However, all these issues (from model interpretability to results that go beyond accuracy) are usually treated by the modern literature as independent perspectives, mostly tackled one at a time. Taking as an example algorithmic fairness (which is also the main case study in our work), while it is of uttermost importance to provide the end users and the content providers with equitable recommendations, it is also important for service providers (e.g., an online platform) to understand *why* the model behind their platform is unfair. Hence, tackling algorithmic fairness in an interpretable way is a central yet under-explored area.

Graph neural networks (GNNs) [19, 24, 37, 48, 50] have proven to be effective in modeling graph data in several domains, such as information retrieval [11], recommender systems [21, 46], natural language processing [45] and user profiling [7, 8, 33, 44]. Counterfactual methods have recently emerged as an effective way to explain the predictions produced by GNN-based models [10, 23, 27, 47]. Moreover, they have been used to guarantee algorithmic fairness in GNN-based models, in various machine learning tasks systems, by manipulating the topological structure [1, 28, 38]. However, to the best of our knowledge, no approach was ever proposed to explain unfairness in GNN-based recommender systems. Filling this research gap goes beyond a simple application of counterfactual explanations methods for GNNs, so as to uncover unfairness in recommender systems. Indeed, the original methods to explain the predictions in GNN-based models are applied to classic graphs, while recommender systems are characterized by a bipartite nature, since they bridge the interactions between two types of entities (nodes), i.e., users and items. This leads to issues in terms of efficiency, uncovered in detail in Section 4. Moreover, existing approaches that explain unfairness in recommender systems, exploit user and item features to characterize the disparities generated by a model. However, these types of features might be challenging to obtain and most recommendation models work with user-item interaction data and do not exploit additional features.

In this work, we propose a shift of paradigm, by presenting a framework, named GNNUERS (GNN-based Unfairness Explainer in Recommender Systems), which perturbs the original bipartite graph in order to generate a set of recommendations that are fair for the end users, with the minimum possible perturbation on the graph. Thanks to our approach, we are able to uncover the user-item interactions leading to user unfairness, thus explaining under which conditions a model generates disparities. Concretely, our

approach is guided by the demographic parity principle, which ensures that all demographic groups receive the same recommendation effectiveness. Under this paradigm, we propose a perturbation mechanism, which alters the user-item interactions; the edges to be perturbed are selected with a loss function that combines two terms: i) minimizing the absolute pair-wise difference across demographic groups, and ii) minimizing the distance between the original adjacency matrix and the perturbed one. Our framework is not only able to systematically explain user unfairness, but also to uncover existing patterns that justify why unfairness was characterized.

Our contributions can be summarized as follows:

- Based on the research gaps existing in the current literature (Section 2), we formulate the problem of explaining unfairness in GNN-based recommender systems (Section 3) and propose a framework to generate counterfactual explanations of the unfairness they propagate (Section 4);
- We validate our proposal on three state-of-the-art models under four data sets, measuring key utility metrics (Section 5), finally discussing the implications of our work for future advances in this research area (Section 6).

2 RELATED WORK

Our study has the primary goal of explaining unfairness in recommender systems based on graph neural networks, which is a naturally multidisciplinary topic. Therefore, we first contextualize our study with respect to the literature on user fairness in recommendation. We then provide an overview on existing methods adopting counterfactual explanations in recommender systems, not necessarily for explaining user unfairness. Finally, we connect our study with similar methods that proposed to perturb user-item interactions in order to optimize a certain recommendation objective.

2.1 User Fairness in Recommendation

Recent work in the recommendation field has shown an increasing attention to issues of unfairness, from both the user and provider perspectives. Most of those focusing on the end users have modeled fairness at a group level, with a primary focus on gender- and age-based demographic groups, and often accompanied unfairness assessments with technical contributions for mitigating it [4, 5, 13–15, 25, 26, 42, 43]. These methods however do not consider algorithmic ways to explain the causes behind the detected disparities, letting mitigation methods attempt to reduce such disparities by optimizing a certain loss function. Although these mitigation methods usually led to a decrease in unfairness, the underlying causes of unfairness remain unclear, consequently preventing researchers from devising more conceptually informed mitigation methods. We particularly observed that it is generally hard to link the mitigation method logic to the underlying aspects causing unfairness in a given domain. Compared to these prior works, GNNUERS not only aims to identify prior user interactions that potentially led the model to provide unfair recommendations, but also investigates the structural characteristics of the perturbed interactions. It follows that GNNUERS makes it possible to derive conceptual insights that support a better understanding of the unfairness phenomenon.

2.2 Counterfactual Explanations in GNNs

GNNs can be directly applied to graphs to provide an easy way to do node-level, edge-level, and graph-level prediction tasks. A notable advantage of GNNs is that they can capture the dependence of graphs via message passing between the nodes of graphs. Unlike standard neural networks, GNNs hence retain a state representing information from their neighbourhood with arbitrary depth.

Recent work put much emphasis on improving interpretability and explainability [3, 9, 17, 18]. Yet, only a few of them addressed explainability about unfairness [12, 16]. With the increasing importance given to both fairness and explainability, these properties have been comprehensively assessed in GNNs as well. Prior work in the machine learning field has proposed to mitigate unfairness in GNNs by manipulating the graph topological structure [1, 28, 38]. However, such methods have never been investigated on bi-partite graphs adopted for recommendation, and their operationalization of fairness and its mitigation could not be applied in our case.

Other efforts were devoted to explaining GNN predictions by leveraging counterfactual techniques [10, 23, 27, 47]. These techniques aim to find the minimal perturbation to the input (graph) data such that the prediction of the GNN changes. Nevertheless, none of them focused on explaining fairness, but merely addressed explanations of individual predictions. Differently from them, GNNUERS leverages counterfactual techniques to find the minimal perturbation to the input (graph) data such that the unfairness of recommendations produced by the GNN-based model is reduced. Our method therefore differs from other existing solutions from several perspectives. First, we target GNNs applied to bi-partite graphs, instead of focusing on more general graphs typically considered in the machine learning field. Second, we adopted a different notion of counterfactuality, which does not only require that the prediction changes, but also that predictions lead to a certain property. Third, we go beyond the mere creation of the counterfactual explanations, and investigate structural properties of the perturbed data.

2.3 Perturbation in GNN Recommendation

First attempts to explain fairness via data perturbation have been made in [16]. That study focused on exposure fairness of the recommended items and applied a counterfactual technique on feature-aware recommender systems. Their optimization was performed with the aim of finding the minimal perturbation to user and item features that can reduce exposure unfairness. It should be noted that these explanations are therefore based on pre-computed user and item features and their importance with respect to exposure unfairness. Conversely, GNNUERS touches on user-item interactions, i.e., the main source of personalization for collaborative filtering models, including the GNN-based ones. While we fully acknowledge the fact that these two aspects are complementary, user-item interactions are generally present in any dataset adopted in a recommendation scenario, whereas user and item features, e.g., those in [16], might not always be extracted from the available data. Moreover, from a structural perspective, the latter depend more on the domain and might not generalize across domains.

Other researchers have investigated to what extent GNN-based recommender systems are sensitive to minimal input data perturbations. For instance, the method proposed in [31] aims to find

the minimal perturbation that causes the highest instability in the recommendations, by analyzing the effect of a perturbation over the graph in a black-box setting. However, such method focuses on decreasing recommendation stability and might be used to merely explain such phenomenon. Conversely, GNNUERS can be adopted to optimize different objectives, including unfairness explanation and recommendation instability. For the sake of scope, our study in this paper explores the former and leaves the latter as a future work.

3 PROBLEM FORMULATION

Our paper aims to explain user unfairness in recommendations generated by graph neural network models. Therefore, we first describe the recommendation scenario from a graph perspective. We then formulate the target task, namely user unfairness explanation. Finally, we introduce the definition of fairness adopted in our study.

3.1 Recommendation Task

In recommendation, the goal of the preference model is typically predicting whether or to what extent an (unseen) item would potentially be of interest for a user. In a common scenario, the model uses past interactions between two main entities, namely users U and items I , to learn preference patterns. Each user $u \in U$ is assumed to have interacted with a certain item $i \in I$ in case they rated, liked, or clicked on such item, depending on the applicative scenario. The set of items I_u a user interacted with is referred to as the u 's profile.

Graphs are structures that represent a set of entities (nodes) and their relations (edges). GNNs operate on graphs to produce representations that can be used in downstream tasks. In our case, user-item interactions can be represented by means of an undirected bipartite graph $G = (U, I, E)$, where E is the set of edges representing the interactions and $U \cup I$, with $n = |U| + |I|$, is the set of vertices. No edge exists between vertices of the same type, i.e., $E = \{(u, i) \mid u \in U, i \in I\}$. The recommendation problem can be then solved by leveraging GNNs. They can be applied to a linking prediction downstream task, to predict potentially interesting links between users U and items I in the bipartite graph G .

Let $f(A, W) \rightarrow \hat{R}$ be any GNN, where A is an $n \times n$ adjacency matrix representing G , W is the learned weight matrix of f , and \hat{R} is an $|U| \times |I|$ user-item relevance matrix, with $\hat{R}_{u,i}$ being the linking probability between user u and item i . In other words, A is the input of f , and f is parameterized by W . To avoid cluttered notation, let f be a standard, one-layer GNN. The function f can be then defined as $f(\text{softmax}[D^{-\frac{1}{2}}AD^{-\frac{1}{2}}], W)$, where $D_{i,j} = \sum_j A_{i,j}$ are entries in the degree matrix D , W is the weight matrix, and g is a combining function depending on the GNN implementation. Given the user-item relevance matrix \hat{R} and a user u , items in I are sorted based on their decreasing relevance in \hat{R}_u , and the top- k items are recommended to user u . We refer to the list of items recommended to a user as Q_u and to the set of all recommended lists as Q .

3.2 Unfairness Explanation Task

Given a GNN defined by f , we aim to explain the reason behind disparate estimates across users in the resulting recommendations. To this end, we decided to adopt counterfactual reasoning techniques [23, 27]. In our context, we assume to model counterfactual explanations according to the user history. More precisely, a set of

interactions, perturbed with respect to the original user-item interactions, represents a counterfactual explanation in case the GNN produces at least one different recommendation to users, when these perturbed interactions are used for training. In our graph-based approach, it means that we aim to generate a perturbed version of the adjacency matrix A , i.e., \tilde{A} , that, once used for training the GNN (instead of A), leads to the recommended lists \tilde{Q} , with $\tilde{Q} - Q \neq \emptyset$.

Under our user unfairness explanation task, we specifically aim to produce a perturbed adjacency matrix \tilde{A} (counterfactual explanation) that leads to the highest fairness across users by means of the lowest number of perturbations on the original adjacency matrix A . Algorithmic unfairness has been operationalized through numerous notions, often dependent on the context and the application [34]. It follows that there is no consensus in the recommender systems community on a gold standard definition to apply. Motivated by its increasingly recognized importance in prior work in top- n recommendation [4, 43], we decided to model fairness according to the notion of demographic parity, denoted with the function $dp(Q) : Q \rightarrow \mathbb{R}$ (the higher it is, the fairer). Our formulation and method is, however, flexible to accommodate other notions of fairness. In the context of recommendations, a model meets demographic parity when the recommendation utility estimates across demographic groups, characterized by a certain sensitive attribute, is not systematically different. Under this demographic parity notion of fairness, our goal becomes to generate a perturbed adjacency matrix \tilde{A} that, once used for training the GNN, results in recommended lists \tilde{Q} whose fairness $dp(\tilde{Q})$ is higher than that observed with the original recommendations $dp(Q)$, constrained to the number of perturbed edges with respect to the original adjacency matrix A . Given a GNN f , we seek to minimize the following objective function:

$$\mathcal{L}(A, \tilde{A}) = \mathcal{L}_{fair}(A, f(\tilde{A}, W)) + \mathcal{L}_{dist}(A, \tilde{A}) \quad (1)$$

where \mathcal{L}_{fair} is the term monitoring fairness, operationalized as the complement of the demographic parity function h , and \mathcal{L}_{dist} is the term controlling the distance between the perturbed adjacency matrix \tilde{A} and the original one A , operationalized with a distance function d . In the next section, we describe the way the perturbed adjacency matrix \tilde{A} is generated and how the objective function is translated into a loss function to be minimized with our approach.

4 GNNUERS

In this section we present GNNUERS, a method able to explain unfairness in graph-based recommenders, solving the problem introduced in Section 3.2. We introduce the methods by its main components: (a) first, the perturbation mechanism of the bi-partite graph, which allows to alter the interactions between users and items, in a differentiable way and, (b) the two loss functions that guide the selection of the edges to be perturbed.

4.1 Bi-partite Graph Perturbation

Our graph perturbation approach is inspired by previous work for GNNs explanation for binary classification on plain graphs [27]. However, since GNNUERS aims to perturb a bi-partite graph generated for recommender systems, presents several differences. In [27]

a perturbation matrix P is populated to then generate the perturbed matrix $\tilde{A} = P \odot A^1$. Optimizing for P can eventually include indices for zero entries in A . While for plain graphs this method results to be efficient, for bi-partite graphs it can be memory inefficient, mainly because it requires to store a perturbation value also for the user-user and item-item links. To overcome these limitations, the perturbation in GNNUERS is optimized through a vector p with size B , which is the number of existing edges in the original graph. Our method is memory efficient, especially under sparse graphs, since it needs to store perturbation values only for non-zero entries of A .

Given our unfairness explanation task, we aim to find the interactions in A that generated unfair recommendations through the original recommendations. To do so, we derive a perturbed matrix \tilde{A} , which would present a reduced level of unfairness. Through a direct mapping from p to A , i.e., $h : \mathbb{N}^{|U|} \times \mathbb{N}^{|I|} \rightarrow \mathbb{N}_{<B}$ (considering p as a 0-indexed vector) function that maps the 2D indices (u, i) of A to a 1D index for p , we can optimize p to replace non-zero entries $A_{u,i} \neq 0$, such that an edge $A_{u,i}$ is deleted if $p_{h(u,i)} = 0$. In other words, the perturbed matrix \tilde{A} is populated as follows:

$$\tilde{A}_{u,i} = \begin{cases} p_{h(u,i)} & \text{if } 0 \leq h(u,i) < B \\ A_{u,i} & \text{otherwise} \end{cases} \quad (2)$$

The perturbation is therefore based by the way h and B are defined.

Following [27, 35], first we define a real valued vector \hat{p} , we apply a sigmoid transformation, and then a binarization of the entries such that values ≥ 0.5 become 1, while values < 0.5 become 0, obtaining eventually p . The initialization of \hat{p} should guarantee $\tilde{A} = A$, i.e., a real-valued α is selected to initialize \hat{p} , such that $p_i = 1, \forall i \in [0, B)$. In all the experiments in Section 5 we set $\alpha = 0$.

4.2 Perturbed Graph Generation

Based on the protocol described above, GNNUERS modifies the adjacency matrix edges by means of the perturbation vector p . The decision process of which perturbed edges will be deleted is performed by the counterfactual model \tilde{f} introduced in (1). \tilde{f} extends the GNN-based recommender system f using p as parameter and the weights W learnt by f as additional input. In detail, $\tilde{f}(A, W; \hat{p}) \rightarrow \tilde{R}$ performs the same steps of the pipeline of f , but using \tilde{A} : \tilde{f} predicts the altered relevance matrix \tilde{R} by combining the normalized version of the perturbed adjacency matrix $\tilde{L} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ and W according to the implementation of the original model f ($\tilde{D}_{u,u} = \sum_i \tilde{A}_{u,i}$). Therefore, \tilde{f} learns only \hat{p} , while the weights W , already optimized by f to maximize the recommendation utility, remain constant.

As explained in Section 4.1, p is generated from \hat{p} , whose values get updated during the learning process. At different steps of the latter, the values of \hat{p} could oscillate close to the threshold that determines if p will be 0 or 1 at the respective indices. Considering aspects such as floating-point errors or dropout layers, the oscillation could negatively affect the update of \hat{p} , due to previously perturbed edges being restored, or vice versa. To counter this phenomenon, the perturbation algorithm is constrained by the usage of a policy that prevents a deleted edge from being restored, such that the number of perturbed edges follows a monotonic trend.

¹ \odot denotes the Hadamard product.

4.3 Loss Function Optimization

The previous section introduced \tilde{f} , the counterfactual model responsible of the generation of the perturbed adjacency matrix \tilde{A} . The optimization of the perturbation vector p is guided by the loss function defined in (1), where \mathcal{L}_{fair} is based on Demographic Parity (DP), the fairness notion described in Section 3.2. Recent works [4, 43] operationalize DP as the mean of the absolute pair-wise utility difference across all demographic groups. Formally:

$$\mathcal{L}_{fair}(\tilde{R}, A) = \frac{1}{\binom{|G|}{2}} \sum_{1 \leq i < j \leq |G|} \left\| S(\tilde{R}^{G_i}, A^{G_i}) - S(\tilde{R}^{G_j}, A^{G_j}) \right\|_2^2 \quad (3)$$

where G is the set of considered demographic groups and S is a function that measures the recommendations utility level.

Following other works proposing methods to face unfairness issues in recommendation [2, 22, 25], we focus on a binary setting, with sensitive attributes comprised of two demographic groups. In the case of $G = \{\text{males}(M), \text{females}(F)\}$, \mathcal{L}_{fair} optimizes to guarantee an equal utility between males and females, i.e.:

$$\left\| S(\tilde{R}^M, A^M) - S(\tilde{R}^F, A^F) \right\|_2^2 = 0$$

In our work, the group with higher utility is denoted as *unprotected* and the one with lower utility as *protected*. This facilitates the reader to better contextualize the approach with respect to fairness.

Normalized Discounted Cumulative Gain (NDCG) was selected as the utility metric S . However, due to the non-differentiability of the sorting operation performed to compute NDCG, we adopt an approximated version [32, 43], which we refer as *NDCGApproxLoss*:

$$\begin{aligned} \text{NDCGApproxLoss}(r, a) = & - \frac{1}{\text{DCG}(a, a)} \sum_i \frac{2^{a_i} - 1}{\log_2(1 + z_i)} \\ \text{s.t. } z_i = & 1 + \sum_{j \neq i} \sigma \left(\frac{r_j - r_i}{\gamma} \right) \end{aligned} \quad (4)$$

where DCG is the Discounted Cumulative Gain, r is the item relevance score produced by the recommender system, and γ is a scaling constant. We fix $\gamma = 0.5$ for the experiments in Section 5.

Any differentiable distance function can be adopted as the distance loss \mathcal{L}_{dist} [27]. In GNNUERS, d is based on the absolute element-wise difference between \tilde{A} and A , defined as follows:

$$\mathcal{L}_{dist} = \beta \frac{1}{2} \sigma \left(\sum_{i,j} \left\| \tilde{A}_{i,j} - A_{i,j} \right\|_2^2 \right) \quad (5)$$

A sigmoid function is used to bound the distance loss to same range of \mathcal{L}_{fair} , i.e., $[0, 1]$. In particular, we adopted $\sigma(x) = |x| / (1 + |x|)$ which needs a significantly higher number of perturbed edges to reach 1 compared to the popular logistic function, hence covering a wider range of values. β is a normalization parameter that balances the losses, due to the trend of \mathcal{L}_{fair} to report values $\ll 0.5$. In all the experiments in Section 5, we set $\beta = 0.01$ as the value that works best to normalize \mathcal{L}_{dist} .

4.4 Gradient Deactivation

The optimization of (4) takes into account the approximate NDCG measured on the predicted recommendations for the protected group and for the unprotected group. The update of the real-valued perturbation vector \hat{p} is then affected from the viewpoint of both

demographic groups. In particular, GNNUERS selects edges that could simultaneously optimize two objectives: increasing utility for the protected group and decreasing it for the unprotected one. However, the edges that are going to be perturbed for one of the objectives could negatively affect the other one, and vice versa. To this end, we perform a gradient *deactivation* on the recommendations generated for the protected group, i.e., the back-propagation updates the perturbation vector only from the unprotected group viewpoint. This procedure is applied only on the protected group, such that GNNUERS objective is to delete edges generating the gap in recommendation utility between unprotected and protected users. Deactivating the gradient does not limit the group of edges that can be perturbed because the optimization does not involve only the user nodes, but also the item ones. Hence, GNNUERS could delete all the edges connected to an item node, both coming from user nodes of the unprotected and protected group. For conciseness, we will use the terms *deactivated* and *activated* to characterize a group associated with inactive and active gradient respectively.

4.5 Resources Usage

In this section, the two steps of the GNNUERS pipeline are examined in terms of memory footprint and execution time complexity. The first step regards the generation of the perturbed matrix \tilde{A} at each step of the learning process by means of (2), which requires to store only the real-valued perturbation vector \hat{p} . Leveraging a sparse representation of A and \tilde{A} , the perturbation time complexity is dependent only on the number of perturbed edges B , i.e., $O(B)$. The second step, that is the optimization process in Sections 4.2-4.3 to learn p , has no memory footprint and is executed for C iterations. Hence, given Θ the execution time for the inference step of the GNN-based recommender system and Ψ the \mathcal{L}_{fair} execution time, $O(\Theta\Psi CB)$ is the time complexity of the perturbed graph generation.

5 EVALUATION

In this section, we examine the GNNUERS explanations with experiments aimed at answering the three research questions:

- **RQ1:** Can the perturbation of the graph topological structure mitigate recommendation utility unfairness under the operationalised fairness notion?
- **RQ2:** Does the edges deletion reduce the unprotected group utility while minimally affecting the protected group one?
- **RQ3:** Does GNNUERS perturbation depend on the topological graph properties of the nodes associated to removed edges?

All the aspects regarding the data manipulation, training and assessment of the GNN-based recommender systems were built upon the framework Recbole [51]. The experiments were ran on a A100 GPU machine with 80GB VRAM and 90GB RAM.

5.1 Graph Topological Properties

GNNUERS generates explanations in the form of user-item interactions that made a GNN-based recommender system generate unfair outcomes. Each edge deleted from the graph unlinks a user and an item node, modifying the network topological structure and affecting the properties characterizing all the nodes, e.g., degree. GNNUERS edges selection process can then be described by the properties of the nodes in the removed edges.

Table 1: Statistics of the four data sets used in our experimental protocol. Repr. stands for Representation, Min. for Minimum. G stands for Gender, A for Age. The line over the graph properties denotes their average

	ML-1M [20]	FENG ²	LFM-1K [6]	INS ³
# Users	6,040	25,741	268	346
# Items	3,706	23,643	51,609	20
# Interactions	1,000,209	708,919	200,586	1,879
Min. User DEG.	20	5	21	5
Domain	Movie	Grocery	Music	Insurance
Repr.	G: F: 28.3%; M: 71.7% A: O: 43.4%; Y: 56.6%	NA O: 45.5%; Y: 54.5%	F: 42.2%; M: 57.8% O: 42.2%; Y: 57.8%	F: 23.4%; M: 76.6% O: 49.4%; Y: 50.6%
User DEG	G: F: 101.8; M: 122.7 A: O: 106.1; Y: 124.9	NA O: 20.7; Y: 19.6	F: 496.7; M: 545.3 O: 657.5; Y: 428.0	F: 4.2; M: 4.5 O: 4.3; Y: 4.5
User SP	G: F: 92.8%; M: 92.7% A: O: 92.9%; Y: 92.6%	NA O: 99.6%; Y: 99.6%	F: 96.1%; M: 96.4% O: 96.5%; Y: 96.1%	F: 59.9%; M: 61.4% O: 60.4%; Y: 61.7%
User RB	G: F: 95.2%; M: 96.1% A: O: 95.2%; Y: 96.4%	NA O: 57.2%; Y: 56.2%	F: 99.0%; M: 98.8% O: 98.9%; Y: 98.8%	F: 97.1%; M: 97.1% O: 97.0%; Y: 97.2%

To this end, we selected three properties that reflect different networks topological aspects and their relation to features examined recommender systems tasks, e.g., popularity bias. Let $z \in Z$ be a generic node of G , i.e., $Z = U$ if z is a user or $Z = V$ if z is an item, the nodes properties are defined as follows:

- **Degree (DEG):** the number of edges connected to each node. For a user node u it represents the history length, i.e., I_u , for item nodes it represents their popularity.
- **Sparsity (SP):** it represents the tendency of a node to be connected to low-degree nodes. For user nodes it represents the tendency to interact with niche items, for item nodes it describes the disinterest of their peers, where users' disinterest is higher as their histories length is shorter. Formally:

$$SP_z = 1 - \frac{\sum_{i=1}^{|I_u|} \frac{|\{z'' \mid (z', z'') \in E \wedge z' \in I_u\}|}{|Z|}}{|I_u|} \quad (6)$$

- **Reachability (RB):** it represents how a node z is close to the other nodes $z' \in Z \setminus \{z\}$. Given the bi-partite nature of recommender systems networks, we consider two users (items) being distant n if the shortest path that connects them include n items (users). Reachability is the average of nodes of the same type normalized by their distance to the considered node. Formally, for a node $z \in Z$:

$$RB_z = \frac{\sum_{n=1}^N \frac{|\{z' \mid \Gamma(z, z') = n\}|}{n}}{|Z|} \quad (7)$$

where Γ measures the shortest path length between two nodes of the same type.

5.2 Data Preparation

Extensive research in user fairness in recommender systems is challenging due to the limited datasets including users' sensitive information. We relied on the artifacts of a recent work accounting unfairness issues in recommendation [4], which performed a fairness assessment on two corpora: MovieLens 1M (ML-1M), on the movie domain, and Last.FM 1K (LFM-1K), on the music domain. We extended the set of datasets by including Insurance (INS), on the insurance domain, and Ta Feng (FENG), on the grocery domain⁴. All datasets include age and gender (except FENG) information for

⁴Yelp [29] was also considered to include the business domain, but the users' gender information was predicted by their name, making questionable analyses on this dataset.

all users and their statistics are listed in Table 1, where the graph properties values regard only the training set.

User nodes in INS and FENG were filtered by their number of interactions, i.e., their degree, so as to take into account users with histories made up of at least 5 items. Duplicated interactions, e.g., users buying the same product twice in FENG, were removed. On the basis of the binary setting mentioned in Section 4.3 and as done in [4], INS and FENG age labels were binarized as *Younger* (*Y*) and *Older* (*O*), such that the *Younger* group is more represented than the *Older* one for consistency with ML-1M and LFM-1K, while gender labels were already binary.

We also adopted the splitting strategy used in [4] for each dataset: per each user, 20% (the most recent if a timestamp is available, randomly sampled otherwise) of the interactions forms the test set; the remaining interactions are split again, such that 10% (selected in the same way) of this interactions subset forms the validation set and the remaining 70% forms the train set. The validation set was used to select the training epoch where the model best performed in terms of recommendation utility on the adjacency matrix *A*. Given the goal of finding the edges causing unfairness in the test set, the truth ground labels T_{test} of the latter were extracted to optimize the fairness loss in (4).

5.3 Models

Recently, novel GNNs have been devised to solve the top-n recommendation task. To leverage such powerful models, we relied on Recbole, which includes different families of GNNs-based recommender systems. GNNUERS was adopted on the following models:

- GCMC [36]: this method is comprised of two components: a graph auto-encoder, which produces a node embedding matrix, and a decoder model, which predicts the relevance of the missing entries in the adjacency matrix from the node embedding matrix.
- NGCF [40]: this state-of-the-art GNN-based recommender system propagates embeddings in the user-item graph structure. In particular, it leverages high-order connectivities in the user-item integration graph, injecting the collaborative signal into the embedding process in an explicit manner.
- LighGCN [21]: it is a simplification of a GCN, including only the most essential components for collaborative filtering, i.e., the neighborhood aggregation. It uses a single embedding as the weighted sum of the user and item embeddings propagated at all layers in the user-item interaction graph.

These three GNNs are trained with the default hyper-parameters defined by Recbole, which provides a specific hyper-parameter configuration for each model.

5.4 Explanation Baseline Methods

As mentioned in Section 4.2, GNNUERS in its base form applies a policy that prevents the algorithm from restoring previously deleted edges. Additionally, we examined an extension of GNNUERS by applying another policy, *Connected Nodes* (*CN*): it limits the perturbation to the edges connected to user nodes of the advantaged demographic group to investigate whether recommendations unfairness is only due to the interactions performed by the advantaged

group. Therefore, *CN* guides the learning process to select the users' actions of the advantaged group that made *f* favor them.

The literature does not include baselines that explain unfairness in the form of user-item edges as GNNUERS. The works proposing unfairness explainability methods in recommendation [12, 16] select relevant user/item features as explanations, which cannot be compared with the ones generated by our framework. Other alternative counterfactual explainability algorithms in GNNs [23, 27] generate explanations at the node level, which cannot be adapted to envision the unfairness task at the network level.

To this end, we introduce *RND-P* as sanity check, a baseline algorithm that at each iteration randomly perturbs edges with a probability ρ , such that it mimics the GNNUERS edges selection process, but based on a random choice. Given the size diversity of our evaluation datasets, we set $\rho = 1/(|E|/10)$ as the value that works best across the selected epochs, such that *RND-P* perturbs edges depending on the network size to prevent this method from deleting all the edges in a few iterations.

The explanations methods were executed on all the models and datasets over 800 epochs adopting an early stopping method when \mathcal{L}_{fair} does not improve with a delta higher than 0.001 for at least 40 consecutive epochs.

5.5 RQ1: Unfairness Explainability Benchmark

We first investigated the capability of GNNUERS to select counterfactual explanations that effectively optimize (4). GNNUERS learning process selects users coming from both demographic groups, stores them in fixed size batches according to their distribution in the dataset and, optimizes the loss to minimize disparity of NDCG@10 (average) between the protected and unprotected group. The evaluation follows an analogous process: we randomly sample 100 subgroups with the same demographic groups distribution, with sample size equal to the batch size. This choice is also due to reduce the sampling bias present in the datasets, i.e. the evaluation is not affected by the different sample size of unprotected and protected groups. We measure the differences with ΔNDCG , which corresponds to the differences in performance between the two user groups. We compare our proposed solution with *RND-P* and the original values of ΔNDCG .

Age. The Figure 1 shows the ΔNDCG distribution across the subgroups by the demographic groups of "age". For each boxplot, on top of it, we include the Student's *t*-test p-value significance of the difference between the means of each distributions pair (it is included IFF the p-values are lower than 0.05). On the bottom of the plots, we include the NDCG@10 after the perturbation and the percentage of deleted edges. First, we can see how in ML-1M and FENG, our methods significantly narrows the NP distribution of the age subgroups ΔNDCG by perturbing just 1% of the edges. On the other hand, the perturbation applied by *RND-P*, in some cases can generate a decrease in unfairness, but removing a larger fraction of edges. Specifically, GNNUERS+CP is the model which with the least perturbations, generate competitive decreases in ΔNDCG . On INS, no explanation method is consistent reducing ΔNDCG . This means that the demographics groups derived from "age" do not impact the differences in performance of the recommenders. Generally, we can see how both GNNUERS and GNNUERS+CP, are able to significantly

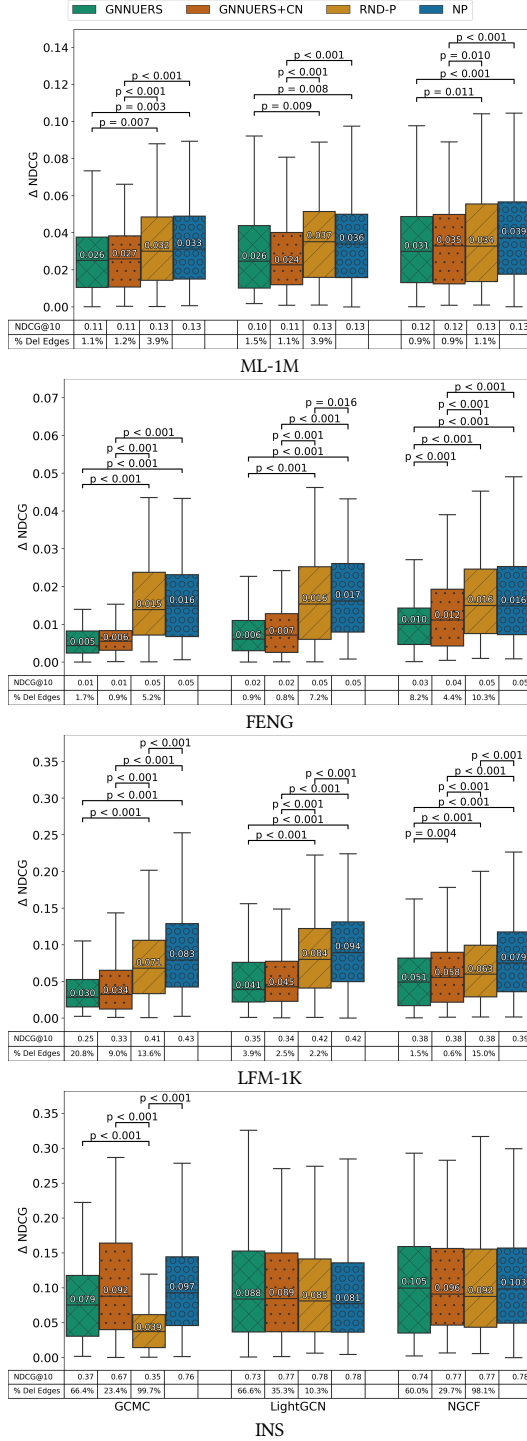


Figure 1: Distribution of ΔNDCG between younger and older users subgroups, randomly sampled 100 times. A Student's t -test is performed between each pair of boxes and the respective p -value is shown if it is lower than 0.05

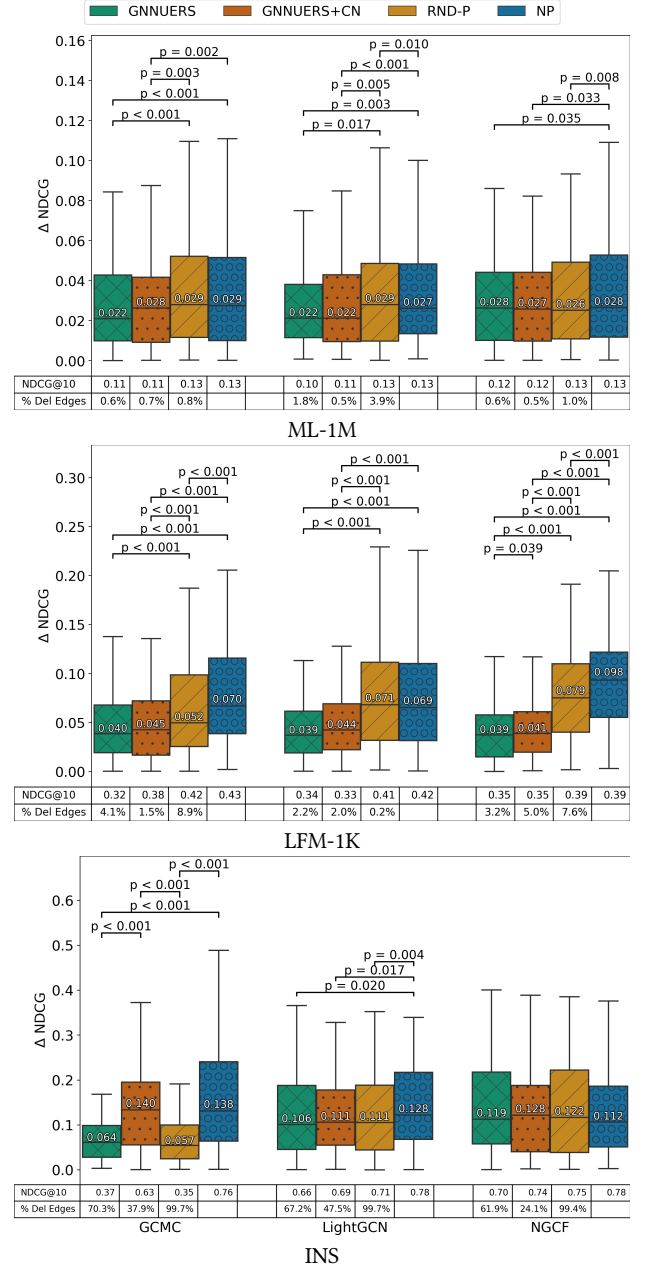


Figure 2: Distribution of ΔNDCG between males and females users subgroups, randomly sampled 100 times. A Student's t -test is performed between each pair of boxes and the respective p -value is shown if it is lower than 0.05

reduce ΔNDCG in most of the settings, selecting small subsets of deleted edges. Interestingly, in some cases, the $\text{NDCG}@10$ drops significantly (LFM-1K and FENG), while in others remain consistent (ML-1M). This means, that our algorithms are able to detect edges which contributes significantly to increase unfairness and then to improve performances for only one subgroup.

Table 2: For both protected and unprotected groups each column include the original value of NDCG and in the brackets, the decrease in NDCG after applying GNNUERS. Unprotected group values are highlighted and in italic.

	Model	Policy	Age		Gender	
			Younger	Older	Males	Females
ML-1M	GCMC	GNNUERS	<i>0.10*</i> (-25.7%)	0.10* (-12.3%)	<i>0.10*</i> (-25.4%)	0.10* (-12.5%)
		GNNUERS+CN	<i>0.11*</i> (-17.0%)	0.11* (-07.1%)	<i>0.10*</i> (-19.8%)	0.11* (-08.3%)
	LightGCN	GNNUERS	<i>0.10*</i> (-23.0%)	0.10* (-14.9%)	<i>0.10*</i> (-23.8%)	0.10* (-17.4%)
		GNNUERS+CN	<i>0.11*</i> (-19.5%)	0.10* (-12.0%)	<i>0.11*</i> (-16.4%)	0.10* (-09.3%)
	NGCF	GNNUERS	<i>0.13*</i> (-06.2%)	0.12* (-01.3%)	<i>0.13*</i> (-05.9%)	0.12* (-00.7%)
		GNNUERS+CN	<i>0.13*</i> (-07.1%)	0.12 (-00.1%)	<i>0.13*</i> (-05.0%)	0.12* (-00.9%)
FENG	GCMC	GNNUERS	0.01* (-69.7%)	<i>0.01*</i> (-83.7%)	-	-
		GNNUERS+CN	0.02* (-61.6%)	<i>0.01*</i> (-76.3%)	-	-
	LightGCN	GNNUERS	0.02* (-49.9%)	<i>0.02*</i> (-64.4%)	-	-
		GNNUERS+CN	0.02* (-47.5%)	<i>0.02*</i> (-64.7%)	-	-
	NGCF	GNNUERS	0.04* (-15.4%)	<i>0.04*</i> (-33.3%)	-	-
		GNNUERS+CN	0.04* (-04.2%)	<i>0.04*</i> (-17.4%)	-	-
LFM-1K	GCMC	GNNUERS	0.26* (-34.3%)	<i>0.24*</i> (-49.4%)	0.32* (-18.8%)	<i>0.31*</i> (-33.3%)
		GNNUERS+CN	0.33* (-14.9%)	<i>0.33*</i> (-29.8%)	0.37* (-08.0%)	<i>0.38*</i> (-17.5%)
	LightGCN	GNNUERS	0.34* (-08.4%)	<i>0.35*</i> (-25.2%)	0.33* (-12.3%)	<i>0.34*</i> (-24.7%)
		GNNUERS+CN	0.33* (-11.2%)	<i>0.34*</i> (-27.8%)	0.33* (-14.0%)	<i>0.33*</i> (-27.3%)
	NGCF	GNNUERS	0.36* (-01.0%)	<i>0.40*</i> (-07.2%)	0.34 (-00.0%)	<i>0.35*</i> (-19.3%)
		GNNUERS+CN	0.36* (-01.7%)	<i>0.41*</i> (-04.9%)	0.34* (-01.8%)	<i>0.36*</i> (-16.7%)
INS	GCMC	GNNUERS	<i>0.37*</i> (-51.5%)	0.36* (-52.7%)	<i>0.36*</i> (-54.0%)	0.39* (-45.1%)
		GNNUERS+CN	<i>0.65*</i> (-14.9%)	0.66* (-12.0%)	<i>0.64*</i> (-19.3%)	0.63* (-10.9%)
	LightGCN	GNNUERS	<i>0.74*</i> (-05.6%)	0.72* (-06.6%)	<i>0.66*</i> (-17.2%)	0.66* (-09.9%)
		GNNUERS+CN	<i>0.76*</i> (-03.3%)	0.77 (-00.7%)	<i>0.68*</i> (-14.3%)	0.70* (-03.3%)
	NGCF	GNNUERS	0.73* (-04.8%)	<i>0.73*</i> (-05.1%)	<i>0.70*</i> (-13.2%)	0.75 (-00.9%)
		GNNUERS+CN	0.76 (-00.7%)	<i>0.77</i> (-00.8%)	<i>0.74*</i> (-08.2%)	0.77 (-02.0%)

Gender. GNNUERS generates a significant decrease in Δ NDCG also for the subgroups generated by the attribute "gender", as shown in Figure 2. In ML-1M and LFM-1K, GNNUERS significantly mitigates unfairness for all the models modifying, the network topologies by an even lower number of edges compared to the same experiments on age groups. Randomly perturbing edges (RND-P) does not decrease Δ NDCG in these cases, while our method has proven to be effective regardless of the sensitive attribute that defines the demographic groups. On INS, differently from what seen before, GNNUERS can reduce unfairness between gender groups, by deleting a relevant lower number of edges compared to RND-P. This result emphasizes how crucial is to select the right demographic attribute affecting the results.

RQ1. *Except extreme cases, GNNUERS selects edges that systematically and significantly explain unfairness, regardless of the data, models and demographic groups on which is applied.*

As shown in both demographic groups, NDCG is not always drastically impacted globally, for this reason we run a more thorough analysis on the impact on utility.

5.6 RQ2: Impact on Recommendation Utility

GNNUERS is devised to minimize the gap in recommendation utility between the demographic groups, without or minimally affecting the utility for the protected group. We empirically evaluate this aspect, by examining the edges deletion impact on the recommendation utility for each demographic group. The NDCG@10 was measured individually for both demographic groups to then averaging it by groups. The impact on recommendation utility was measured as the change in utility after applying the perturbation. To estimate the significance of this change, a Student's t -test was

performed between the 100 NDCG@10 averages measured on the recommendations altered from each explanation method and the ones generated from the non-perturbed network. For this analysis we consider GNNUERS and its extended version GNNUERS+CN.

The Table 2 shows: the average utility (highlighted for the unprotected group); seen between brackets, the utility change after perturbing the edges; for each value, the symbol (*) to denote the significance of the statistical test with the 95% of confidence interval. We can see how, for any dataset and model, the NDCG change for the unprotected group is greater than the protected group one. This confirms that our algorithms can select the edges responsible for an higher utility for the unprotected group. However, also the NDCG for the protected group is affected in most of the experiments. This is because the results are model dependent and removing edges, reduces the connectivity and then the information propagation through the GNN. Also, higher NDCG losses for the unprotected groups reflect a better unfairness mitigation, as seen for GCMC in FENG⁵. Based on this observation, since GNNUERS perturbations for NGCF result in the lowest unfairness mitigation w.r.t. the other models in the previous RQ, for the same model it reports the lowest loss in utility for both demographic groups. As a matter of fact, not only for NGCF the NDCG for the protected group is minimally affected, but it also increases in some settings, e.g., for younger users in LFM-1K. The GNNUERS+CN policy enforces this behavior more than GNNUERS by a slight amount, except for FENG, where the NDCG is equal between demographic groups, but GNNUERS report an additional 10% loss in utility. Using GNNUERS+CN edges selection is then beneficial to reduce the impact on the NDCG for the protected group.

RQ2. *GNNUERS and GNNUERS+CN reduce the utility for unprotected group, detecting edges that generated a disparity in performance, while reporting a negligible loss for the protected one.*

5.7 RQ3: Edges Selection Process

The findings of the first two research questions regard the GNNUERS unfairness explainability and its impact on the recommendation utility of each demographic group. However, we have no information about the logic followed by GNNUERS to select an edge over another and whether this selection process depends on topological properties of the networks. To this purpose, we look for a clarification by analyzing the dependency of the number of perturbed edges from the network topological properties defined in Section 5.1. This dependency was estimated by Kullback-Leibler (KL) Divergence, due to its asymmetrical nature and its ability to reveal the information "difference" between two distributions, where a lower value denote a higher dependency. It was preferred over simpler correlation measures, e.g., Pearson's r , under the hypothesis that the GNNUERS edges selection depends on high-order interconnections between nodes, which cannot be uncovered by a mere linear relationship over the network properties. KL divergence is affected by the sample size, i.e., the number of nodes, so dependency measures on big networks are higher compared to smaller ones, but lower values still denote higher dependency under the

⁵GNNUERS learning process could be stopped once a desired level of fairness or utility is reached, depending on the application requirements.

Table 3: Dependency of number of deleted edges distribution from graph properties distribution estimated with KL divergence. Graph properties are measured on user (Y: Younger, O: Older) and item (I: Item) nodes. Lower values denote higher dependency. Unprotected group values are highlighted and in italic.

	ML-1M						FENG						LFM-1K						ML-1M						FENG						LFM-1K					
	DEG		SP		RB		DEG		SP		RB		DEG		SP		RB		DEG		SP		RB		DEG		SP		RB		DEG		SP		RB	
	Y	O	Y	O	Y	O	Y	O	Y	O	Y	O	Y	O	Y	O	Y	O	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I			
GCMC	2.90	4.41	1.39	2.89	1.40	2.87	2.79	2.44	2.57	2.33	2.50	2.26	0.08	0.18	0.30	0.20	0.30	0.19	1.12	2.98	2.81	2.78	7.11	6.62	1.27	4.50	3.88									
LightGCN	3.31	3.77	1.46	1.95	1.47	1.95	4.07	3.44	3.15	2.82	3.12	2.78	0.46	0.75	0.21	0.51	0.20	0.51	2.83	5.85	5.65	4.01	8.28	7.78	3.72	7.27	6.58									
NGCF	3.37	4.64	1.92	3.41	1.95	3.41	1.13	1.01	1.08	1.26	0.99	1.16	0.32	0.46	0.20	0.22	0.19	0.22	1.32	2.61	2.45	1.67	5.50	5.06	4.25	8.70	7.92									

same dataset. Dependency is not examined on INS because all the values are close to each other due to the low number of nodes in the dataset. GNNUERS+CN was not included in the following experiments because the edges deletion constraint could only reflect a dependency from the user nodes of the advantaged group. Dependency from user nodes graph properties is analyzed individually only for age groups, due to similar observations w.r.t. gender groups.

The KL divergence reported in Table 3 is measured between the distribution of the number of deleted edges and the graph properties distribution of user (Y: Younger, O: Older) and item (I: Item) nodes. Considering at first only the user node properties, the number of deleted edges is more dependent from the unprotected group nodes properties in some settings, but in others the KL divergence is lower for the protected group nodes. In particular, GNNUERS perturbations on NGCF are more dependent on the protected group nodes properties in two of the datasets where the same group was minimally affected by this model, i.e., FENG and LFM-1K. Hence, GNNUERS on NGCF selects the edges that reduce the utility for the unprotected group depending on the protected group nodes properties SP and RB. A similar behavior is reported for LightGCN on LFM-1K, where, as NGCF, the utility loss for the protected group is much lower than the unprotected group one. Across properties, GNNUERS perturbations systematically depend more on user nodes SP and RB than DEG for all the models and datasets (except for NCGF on FENG), regardless of the demographic group. The capacity to reach user nodes from other ones and the users' tendency to interact with niche items are relevant factors for GNNUERS deletion process.

Drawing the attention towards the item nodes properties, it is clear from the KL divergence that GNNUERS is much more dependent on item nodes DEG, i.e., the items popularity, than the other properties. This observation relates to the bias popularity issue in recommender systems, which is clearly taken into account by our framework. Users interacting with popular items in the test set are probably those same users receiving high-utility recommendations, which include the popular items. GNNUERS edges perturbation focuses on high-degree item nodes, such that the relative items positions in recommendation lists change and eventually do not get included in top- n lists.

Combining the observations from user and item nodes, we hypothesize a common scenario where unfairness is related to popularity bias and depicted by the GNNUERS perturbation dependency from user nodes SP, RB and item nodes DEG. Such scenario would consider the edges linking highly reachable user nodes characterized by low sparsity, i.e. high tendency to interact with popular items, with popular items, explainable by GNNUERS.

RQ3. GNNUERS edges perturbation process can be described by the dependency from user nodes SP, RB, and item nodes DEG.

6 DISCUSSIONS AND LIMITATIONS

In this paper, we first depicted the state of the art on research into analysis and mitigation of unfairness in recommendation systems, into techniques that manipulate the network structure to improve fairness or explain predictions of GNNs (Section 2). We presented GNNUERS, a framework that learns the set of edges causing unfairness in GNNs recommendations (Section 4). We adopted our method on three state-of-the-art GNN-based recommender systems and evaluated it on four real-world datasets to explain recommendation utility unfairness between gender groups and between age groups (Section 5). In this section, we connect the findings of our experiments, present the GNNUERS limitations and future extensions.

GNNUERS is the first step towards explaining unfairness in recommender systems in terms of user actions, hence finding what the users of the unprotected group did to make the model favor them. GNN-based recommender systems generate fairer recommendations once the edges selected by our framework are removed from the network used for inference. Our results show that edges selected by GNNUERS mitigate unfairness in all the settings, except for extreme cases, e.g. low number of items in INS. The GNNUERS objective function reduces the recommendation utility disparity negatively affecting the unprotected group utility more than the protected one, and, under certain conditions, increasing it for the latter group. GNNUERS edges selection process is dependent on user nodes SP, RB and by item nodes DEG, which could describe common recommendation phenomena, e.g. popularity bias.

Our findings are based on experiments on four data sets, each one coming from a different domain and covering a wide range of scenarios, e.g. high and low sparsity, high and low number of interactions. The size of the selected datasets could still not be enough to guarantee the generalizability of our conclusions, especially for INS, whose edges do not cover neither 1% of the other networks. However, the datasets comprising sensitive information in recommender systems literature are still limited. Other than that, even though GNNUERS works with sparse representation of adjacency matrices, working with GNNs requires high amount of memory and computational power. This can impede the use of the few datasets that provide users' protected attributes and have a greater size than the ones used in our work, e.g. Last.FM 2B [30], BookCrossing [52].

GNNUERS was used to explain recommendation utility unfairness between two demographic groups for two sensitive attributes. The adopted binary setting better highlights the edges selection performed by our framework and offers a clearer overview of the

GNNUERS explainability power. Even if we considered protected attributes like gender and age as binary features, they are by no means binary and the age labels binarization was performed according only to the desired final groups distribution, without discriminative choices against users' age. Nonetheless, GNNUERS could be extended to generate unfairness explanations for an higher number of demographic groups and sensitive attributes at once.

Future works will extend the evaluation of GNNUERS on other GNN-based recommender systems, other datasets to better profile the logic behind the algorithm. Other sensitive attributes and a larger number of demographic groups will also be considered to examine the generated explanations under different scenarios.

REFERENCES

- [1] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. 2021. Towards a unified framework for fair and stable graph representation learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021 (Proceedings of Machine Learning Research, Vol. 161)*, Cassio P. de Campos, Marloes H. Maathuis, and Erik Quaeghebeur (Eds.). AUAI Press, 2114–2124. <https://proceedings.mlr.press/v161/agarwal21b.html>
- [2] Ashwathy Ashokan and Christian Haas. 2021. Fairness metrics and bias mitigation strategies for rating predictions. *Inf. Process. Manag.* 58, 5 (2021), 102646. <https://doi.org/10.1016/j.ipm.2021.102646>
- [3] Giacomo Balloccu, Ludovico Boratto, Gianni Fenu, and Mirko Marras. 2022. Post Processing Recommender Systems with Knowledge Graphs for Recency, Popularity, and Diversity of Explanations. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 646–656. <https://doi.org/10.1145/3477495.3532041>
- [4] Ludovico Boratto, Gianni Fenu, Mirko Marras, and Giacomo Medda. 2022. Consumer Fairness in Recommender Systems: Contextualizing Definitions and Mitigations. In *Advances in Information Retrieval*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer International Publishing, Cham, 552–566.
- [5] Robin Burke, Nasim Sonboli, and Aldo Ordóñez-Gauger. 2018. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA (Proceedings of Machine Learning Research, Vol. 81)*. PMLR, 202–214. <http://proceedings.mlr.press/v81/burke18a.html>
- [6] Óscar Celma. 2010. *Music Recommendation and Discovery - The Long Tail, Long Tail, and Long Play in the Digital Music Space*. Springer. <https://doi.org/10.1007/978-3-642-13287-2>
- [7] Weijian Chen, Fuli Feng, Qifan Wang, Xiangnan He, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. CatGCN: Graph Convolutional Networks with Categorical Node Features. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [8] Weijian Chen, Yulong Gu, Zhaochun Ren, Xiangnan He, Hongtao Xie, Tong Guo, Dawei Yin, and Yongdong Zhang. 2019. Semi-supervised user profiling with heterogeneous graph attention networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2116–2122.
- [9] Xu Chen, Yongfeng Zhang, and Ji-Rong Wen. 2022. Measuring “Why” in Recommender Systems: a Comprehensive Survey on the Evaluation of Explainable Recommendation. *CoRR* abs/2202.06466 (2022). [arXiv:2202.06466](https://arxiv.org/abs/2202.06466)
- [10] Ziheng Chen, Fabrizio Silvestri, Jia Wang, Yongfeng Zhang, Zhenhua Huang, Hongshik Ahn, and Gabriele Tolomei. 2022. GREASE: Generate Factual and Counterfactual Explanations for GNN-based Recommendations. *CoRR* abs/2208.04222 (2022). <https://doi.org/10.48550/arXiv.2208.04222>
- [11] Hejie Cui, Jiaying Lu, Yao Ge, and Carl Yang. 2022. How Can Graph Neural Networks Help Document Retrieval: A Case Study on CORD19 with Concept Map Generation. In *European Conference on Information Retrieval*. Springer, 75–83.
- [12] Yashar Deldjoo, Alejandro Bellogín, and Tommaso Di Noia. 2021. Explaining recommender systems fairness and accuracy through the lens of data characteristics. *Inf. Process. Manag.* 58, 5 (2021), 102662. <https://doi.org/10.1016/j.ipm.2021.102662>
- [13] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2022. Fairness in Information Access Systems. *Found. Trends Inf. Retr.* 16, 1-2 (2022), 1–177. <https://doi.org/10.1561/15000000079>
- [14] Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. 2018. All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness. In *Conference on Fairness, Accountability and Transparency, FAT 2018*, Vol. 81. PMLR, 172–186. <http://proceedings.mlr.press/v81/ekstrand18b.html>
- [15] Gabriel Frisch, Jean-Benoist Léger, and Yves Grandvalet. 2021. Stereotype-aware collaborative filtering. In *Proceedings of the 16th Conference on Computer Science and Intelligence Systems, Online, September 2-5, 2021*, Maria Ganzha, Leszek A. Maciaszek, Marcin Paprzycki, and Dominik Slezak (Eds.). 69–79. <https://doi.org/10.15439/2021F117>
- [16] Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, and Yongfeng Zhang. 2022. Explainable Fairness in Recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 681–691. <https://doi.org/10.1145/3477495.3531973>
- [17] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 196–204. <https://doi.org/10.1145/3336191.3371824>
- [18] Azin Ghazimatin, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. ELIXIR: Learning from User Feedback on Explanations to Improve Recommender Models. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 3850–3860. <https://doi.org/10.1145/3442381.3449848>
- [19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).
- [20] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19. <https://doi.org/10.1145/2827872>
- [21] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 639–648. <https://doi.org/10.1145/3397271.3401063>
- [22] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2018. Recommendation Independence. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA (Proceedings of Machine Learning Research, Vol. 81)*. PMLR, 187–201. <http://proceedings.mlr.press/v81/kamishima18a.html>
- [23] Bo Kang, Jeffrey Lijffijt, and Tijl De Bie. 2021. Explanations for Network Embedding-Based Link Predictions. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases - International Workshops of ECML PKDD 2021, Virtual Event, September 13-17, 2021, Proceedings, Part I (Communications in Computer and Information Science, Vol. 1524)*, Michael Kamp, Irena Koprinska, Adrien Bibal, Tassadit Bouadi, Benoît Frénay, Luis Galárraga, José Oramas, Linara Adilova, Yamuna Krishnamurthy, Bo Kang, Christine Largeron, Jeffrey Lijffijt, Tiphaine Viard, Pascal Welke, Massimiliano Ruocco, Erlend Aune, Claudio Gallicchio, Gregor Schiele, Franz Pernkopf, Michaela Blott, Holger Fröning, Günther Schindler, Riccardo Guidotti, Anna Monreale, Salvatore Rinzivillo, Przemysław Biecek, Eirini Ntoutsi, Mykola Pechenizkiy, Bodo Rosenhahn, Christopher L. Buckley, Daniela Cialfi, Pablo Lanillos, Maxwell Ramstead, Tim Verbeelen, Pedro M. Ferreira, Giuseppina Andresini, Donato Malerba, Ibéria Medeiros, Philippe Fournier-Viger, M. Saqib Nawaz, Sebastián Ventura, Meng Sun, Min Zhou, Valerio Bitetta, Ilaria Bordino, Andrea Ferretti, Francesco Gullo, Giovanni Ponti, Lorenzo Severini, Rita P. Ribeiro, João Gama, Ricard Gavaldà, Lee A. D. Cooper, Naghmeh Ghazaleh, Jonas Richiardi, Damian Roqueiro, Diego Saldana Miranda, Konstantinos Sechidis, and Guilherme Graça (Eds.). Springer, 473–488. https://doi.org/10.1007/978-3-030-93736-2_36
- [24] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*.
- [25] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. In *WWW '21: The Web Conference 2021. ACM / IW3C2*, 624–632. <https://doi.org/10.1145/3442381.3449866>
- [26] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards Personalized Fairness based on Causal Notion. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 1054–1063. <https://doi.org/10.1145/3404835.3462966>

- [27] Ana Lucic, Maartje A. ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. 2022. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 151)*, Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). PMLR, 4499–4511. <https://proceedings.mlr.press/v151/lucic22a.html>
- [28] Jing Ma, Ruocheng Guo, Mengting Wan, Longqi Yang, Aidong Zhang, and Jundong Li. 2022. Learning Fair Node Representations with Graph Counterfactual Fairness. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (Eds.). ACM, 695–703. <https://doi.org/10.1145/3488560.3498391>
- [29] Masoud Mansoury, Bamshad Mobasher, Robin Burke, and Mykola Pechenizkiy. 2019. Bias Disparity in Collaborative Recommendation: Algorithmic Evaluation and Comparison. In *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 20, 2019 (CEUR Workshop Proceedings, Vol. 2440)*, Robin Burke, Himan Abdollahpour, Edward C. Malthouse, K. P. Thai, and Yongfeng Zhang (Eds.). CEUR-WS.org. <http://ceur-ws.org/Vol-2440/paper6.pdf>
- [30] Alessandro B. Melchiorre, David Rekabsaz, Emilia Parada-Cabaleiro, Stefan Brandl, Oleg Lesota, and Markus Schedl. 2021. Investigating gender fairness of recommendation algorithms in the music domain. *Inf. Process. Manag.* 58, 5 (2021), 102666. <https://doi.org/10.1016/j.ipm.2021.102666>
- [31] Sejoon Oh, Berk Ustun, Julian J. McAuley, and Srikanth Kumar. 2022. Rank List Sensitivity of Recommender Systems to Interaction Perturbations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, 1584–1594. <https://doi.org/10.1145/3511808.3557425>
- [32] Tao Qin, Tie-Yan Liu, and Hang Li. 2010. A general approximation framework for direct optimization of information retrieval measures. *Inf. Retr.* 13, 4 (2010), 375–397. <https://doi.org/10.1007/s10791-009-9124-x>
- [33] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2018. Semi-supervised User Geolocation via Graph Convolutional Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2009–2019.
- [34] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 2219–2228. <https://doi.org/10.1145/3219819.3220088>
- [35] Suraj Srinivas, Akshayvarun Subramanya, and R. Venkatesh Babu. 2017. Training Sparse Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 455–462. <https://doi.org/10.1109/CVPRW.2017.61>
- [36] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *CoRR abs/1706.02263* (2017). [arXiv:1706.02263](http://arxiv.org/abs/1706.02263)
- [37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. In *International Conference on Learning Representations*.
- [38] Nan Wang, Lu Lin, Jundong Li, and Hongning Wang. 2022. Unbiased Graph Embedding with Biased Graph Observations. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 1423–1433. <https://doi.org/10.1145/3485447.3512189>
- [39] Shoujin Wang, Xiuzhen Zhang, Yan Wang, Huan Liu, and Francesco Ricci. 2022. Trustworthy Recommender Systems. *CoRR abs/2208.06265* (2022).
- [40] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 165–174. <https://doi.org/10.1145/3331184.3331267>
- [41] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2022. A Survey on the Fairness of Recommender Systems. *ACM Trans. Inf. Syst.* (jul 2022). <https://doi.org/10.1145/3547333> Just Accepted.
- [42] Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. 2021. Fairness-aware News Recommendation with Decomposed Adversarial Learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 4462–4469. <https://ojs.aaai.org/index.php/AAAI/article/view/16573>
- [43] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. 2022. A Multi-Objective Optimization Framework for Multi-Stakeholder Fairness-Aware Recommendation. *ACM Trans. Inf. Syst.* (aug 2022). <https://doi.org/10.1145/3564285> Just Accepted.
- [44] Qilong Yan, Yufeng Zhang, Qiang Liu, Shu Wu, and Liang Wang. 2021. Relation-aware Heterogeneous Graph for User Profiling. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, New York, NY, USA, 3573–3577.
- [45] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7370–7377.
- [46] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.
- [47] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 430–438. <https://doi.org/10.1145/3394486.3403085>
- [48] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.
- [49] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *Found. Trends Inf. Retr.* 14, 1 (mar 2020), 1–101. <https://doi.org/10.1561/15000000066>
- [50] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2022. Deep Learning on Graphs: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 34, 1 (2022), 249–270.
- [51] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 4653–4664. <https://doi.org/10.1145/3459637.3482016>
- [52] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, Allan Ellis and Tatsuya Hagino (Eds.). ACM, 22–32. <https://doi.org/10.1145/1060745.1060754>

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009