2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Equipping Recommender Systems with Individual Fairness via Second-order Proximity Embedding

Kun Wu, Jacob Erickson, Wendy Hui Wang, Yue Ning
Department of Computer Science
Stevens Institute of Technology
{kwu14, jericks2, hwang4, yue.ning}@stevens.edu

***Abstract*—Graph neural networks (GNNs) have been widely used for recommender systems over knowledge graphs. An important issue of GNN-based recommender systems is *individual user fairness in recommendations* (i.e., similar users should be treated similarly by the systems). In this paper, we make the following contributions to enable recommender systems to be equipped with individual user fairness. First, we define new similarity metrics for individual fairness, where these metrics take knowledge graphs into consideration by incorporating both *first-order proximity in direct user-item interactions* and *second-order proximity in knowledge graphs*. Second, we design a novel graph neural network (GNN) named SKIPHOP for fair recommendations over knowledge graphs. By passing latent representations from both first-order and second-order neighbors at every message passing step, SKIPHOP learns user embeddings that capture their latent interests present in the second-order networks. Furthermore, to realize individual user fairness, we add fairness as a regularization to the loss function of recommendation models. Finally, through experiments on two real-world datasets, we demonstrate the effectiveness of SKIPHOP in terms of fairness and recommendation accuracy.**

***Index Terms*—Algorithmic fairness, Recommender systems, Second-order proximity embedding, Graph neural networks**

I. INTRODUCTION

An important issue in recommender systems is that recommendations can be discriminatory against particular groups as well as individual users/items [1], [2]. To address these vital fairness issues, quite a few fairness-aware recommender systems have been designed recently [3], [4], [5]. Despite their efforts on developing new fairness notions and algorithms, most of these previous efforts mainly focus on *group fairness*, i.e., different user groups should be treated equally by recommender systems [3], [4]. However, it has been shown that group fairness is inadequate, as it might be unfair to some individuals within a group [6]. This introduces the complementary notion of *individual fairness*. At a high level, the state-of-the-art notion of individual fairness [6] requires that similar individuals should be treated similarly, where similarity is measured by a task-specific metric. Individual fairness has been studied in knowledge graph (KG) based recommender systems [5]. However, the existing work does not quantify user similarity based on historical interactions. Instead, they only consider users' similarities formulated on the paths connecting user-item pairs for the top-K items. In this paper, we design a new efficient user similarity metric and create an individual fairness-aware algorithm for KG-based recommender systems.

There are several challenges in solving individual unfairness on Knowledge Graph (KG)-based recommendation approaches. First, most of them (e.g., KGAT [7]) utilize multi-layer message passing to collect high-order proximity information in user-item interactions or knowledge graphs when learning embeddings for recommendations. However, multi-layer message passing methods can lose important information and thus limit the representational capacity of the model [8]. Second, although the individual fairness notion [6] is highly appealing, it critically depends on the extent to which the similarity metrics faithfully represent the needs of specific task settings. In the setting of recommender systems, while individual fairness requires that "similar" users should receive "similar" recommendations, how to define the similarity metrics of both users and recommendations remains unclear. Therefore, the most significant barrier to implementing individual fairness in recommender systems, is the construction of similarity metrics for users and recommendations respectively. In particular, since users who have different historical interactions might still have similar interests if the interacted items of these users share the same features, similarity metrics for both users and recommendations should facilitate the information beyond the first-order proximity.

Our contributions. To address these three challenges, we design a new KG-based recommendation algorithm that provides individual fairness to users. We summarize our main contributions as follows. First, we define new similarity metrics for users and items respectively. Based on both similarity metrics, we define individual fairness which requires that similar users (in both historical item interactions and latent interests) should be recommended with items of similar embeddings. Second, we design a novel graph neural network named SKIPHOP that explicitly models both first-order (i.e., user-item interactions) and second-order network structures (i.e., user latent interests) of user nodes in one-layer message passing. The goal of considering second-order proximity is to capture users' latent features in a knowledge graph, so that users who have not only similar historical item interactions but also similar latent interests (e.g., preferences of movie genres) have similar embeddings. Third, we conduct extensive experiments to demonstrate the effectiveness of SKIPHOP compared with three state-of-the-art recommender systems.

IEEE/ACM ASONAM 2022, November 10-13, 2022
978-1-6654-5661-6/22/$31.00 © 2022 IEEE

II. Related Work

Individual fairness in machine learning. The existing fairness definitions and measurements can be grouped into two categories: *group fairness* which is concerned with protected groups (such as gender groups) and requires that some statistic of interest be approximately equalized across groups [9], [10], [11], and *individual fairness* [6] which prevents discrimination against individuals. In this work, we mainly focus on individual fairness. One challenge in defining individual fairness is to pick the appropriate similarity metrics. The similarity metrics should be context-dependent [6], [12]. Recognizing the challenge of defining similarity metrics, we are inspired by InFoRm proposed by Kang *et al.* [13], and choose to use the Jaccard similarity and the JS divergence for measuring user similarity and cosine distance for measuring recommendation similarity for our purposes.

User-side fairness in recommender systems. Group fairness in recommender systems have received much attention from the research community recently [5], [14]. The studies of individual fairness in recommender systems remain relatively limited. Biega *et al.* [15] consider item-side individual fairness. Their fairness notion primarily concerns the level of attention that an individual item receives in proportion to its relevance (ranked position). There are several works on user-side individual fairness, [5], [16], [17], [18]. But, they either utilize Envy-free fairness or counterfactual fairness. The most similar work to us is [5] which quantifies the disparity of users by measuring the average of the pairwise difference in patterns of paths between users and the top-K recommended items. This is fundamentally different from our measurement of individual fairness.

III. Problem Definition

A natural adoption of individual fairness to recommender systems is to require that *any two similar users must receive similar recommendations*. We first define user similarity and recommendation similarity. Then we formally define individual fairness in recommendations.

User similarity. User similarity can be measured in different ways under various contexts. In this paper, we consider two orders of graph structures that would affect the similarity between pairs of users. Given two users u and v, we calculate the following two types of similarity: **(1) Direct interaction similarity (first-order proximity)** that considers the similarity of nodes that are directly connected to u and v such as historical user-item interactions; **(2) Knowledge graph enhanced similarity (second-order proximity)** that measures the similarity of the nodes that are at distance two to u and v. Second-order proximity aims to capture the similarity of latent interests of users, which is the side information provided by a knowledge graph. In this paper, we adopt *Jaccard similarity* to measure the direct interaction similarity: $J_{u,v} = \frac{|H_u \cap H_v|}{|H_u \cup H_v|}$, where H_u is the set of items that user u has directly connected to in the given graph $\mathcal{G}$. We choose Jaccard given its simplicity, but other similarity metrics (e.g., Dice coefficient and overlap coefficient) can also be applied.

For the second-order proximity, we measure the *distribution distance* of latent user features. Specifically, for any two users, we first calculate the distributions of specific auxiliary features of these users in the knowledge graph. We utilize Jensen-Shannon divergence (JS divergence) [19] to calculate the distance between these two distributions. Formally, given two users u and v, and a hidden feature a (e.g., genres or creators) in the second-order proximity of u and v, the distributional similarity of u and v on a feature a is computed as follows:

$$D^a_{u,v} = 1 - \big(\frac{1}{2}\mathrm{KL}(G^a_u||G^a_v) + \frac{1}{2}\mathrm{KL}(G^a_v||G^a_u)\big), \quad (1)$$

where $\mathrm{KL}(P||Q)$ denotes the Kullback–Leibler (KL) divergence between two distributions P and Q, and G^a_u denotes the distribution of feature a of items that user u has explicitly interacted with. Since JS divergence reflects the distance between two distributions, we convert it to the similarity of two distributions by using (1-JS).

Based on the 1st-order and 2nd-order proximity, we measure the *similarity between any two users u and v* as follows:

$$\mathbf{S}_{u,v} = \gamma J_{u,v} + (1-\gamma) D^a_{u,v}, \quad (2)$$

where $J_{u,v}$ and $D^a_{u,v}$ follow Jaccard similarity and Eq. (1) respectively. For simplicity, in this paper, we only consider one feature (e.g. $a =$ genre) when calculating user similarity. However, Eq. (2) can be easily extended to multiple features by averaging $D^a_{u,v}$ for different a.

Recommendation similarity. Given two users u and v, and their recommendations R_u and R_v which are two ranked lists of items, we define recommendation dissimilarity (distance) between two lists of items. For any two items p, q, we measure their dissimilarity as the distance between their embeddings $\boldsymbol{h}_p$ and $\boldsymbol{h}_q$. Formally, given two users u and v, and their top-k recommendations R_u and R_v, the *dissimilarity between R_u and R_v* is measured as follows:

$$\delta_{u,v} = \frac{1}{k^2} \sum_{p \in R_u} \sum_{q \in R_v} d(\boldsymbol{h}_p, \boldsymbol{h}_q), \quad (3)$$

where $\boldsymbol{h}_p$ and $\boldsymbol{h}_p$ are the embeddings of items p and q, and $d(\cdot)$ is a function that measures the distance between the embeddings $\boldsymbol{h}_p$ and $\boldsymbol{h}_q$ such as cosine distance. We choose cosine distance over Euclidean distance and generalized Minkowski distance because of its flexibility and its bounded value domain. Equation (3) evaluates the average pairwise item distance for all items in two recommendation lists. Intuitively, the recommendations that contain similar items will have a low distance. It is worth mentioning that we do not consider using ranked similarity methods due to the computational complexity since we use it as a regularization term in the loss function.

Individual fairness in recommender systems. Based on the definition of user similarity and recommendation dissimilarity, we define *unfairness score* (UF) formally. Given a set of users $\mathcal{U}$, let R_u be the recommendations for user $u \in \mathcal{U}$, the UF

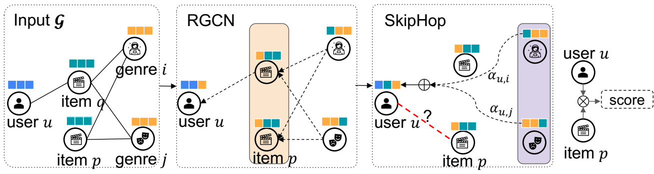


Fig. 1: The framework of SKIPHOP GNN. Orange and purple boxes denote first-order and second-order neighborhood of user u respectively.

score of all the recommendations made by a recommender system, $\{R_u | \forall u \in \mathcal{U}\}$, is computed as:

$$\text{UF} = \log\Big(\sum_{\forall u,v \in \mathcal{U}} \mathbf{S}_{u,v} \times \delta_{u,v} \Big) \text{ if } \mathbf{S}_{u,v} > \tau, \quad (4)$$

where τ is the threshold for selecting similar users. Inspired by InFoRm [13], we set $\tau = \text{mean}(s) + \text{std}(S)$ which denotes a value above average. Unlike [5] that measures individual fairness as the utility disparity between user pairs, our UF score takes both the distance of item embeddings and user similarity into consideration. Our goal is to design a KG-based recommender system that minimizes the UF score while preserving the accuracy of recommendations.

IV. METHODOLOGY

One of the underlying reasons that the conventional RGCN based recommender systems fail to provide individual fairness is that they do not explicitly take second-order neighborhood similarity into consideration, and thus their recommendations do not respect such user similarity. To mitigate this problem, we design a new GNN model named SKIPHOP that provides individual fairness to recommendations. Figure 1 illustrates the overview of SKIPHOP. It consists of three steps of each iteration: *Step 1: Generate RGCN embeddings*: The embedding vectors of the nodes in a given knowledge graph are generated by a conventional RGCN model. *Step 2: Generate* SKIPHOP *embeddings*: With the RGCN embeddings as the input, SkipHop GNN generates embeddings of user nodes that capture users' latent interests by utilizing their second-order proximity in the given KG; *Step 3: Fairness-aware recommendations*: The model learns recommendations from SkipHop embeddings with individual fairness equipped as a regularization term to the loss function. Step 1 simply utilizes the existing RGCN models to generate the embeddings. Therefore, we will present the details of Steps 2 & 3.

A. Learning SKIPHOP Embeddings

When generating node embeddings, SKIPHOP considers second-order message passing, where the nodes receive latent representations from their immediate (first-order) neighbors as well as second-order neighbors at every message passing step. Specifically, SKIPHOP takes the embeddings of user nodes generated by the RGCN model as input, and incorporates these embedding vectors with second-order neighborhoods directly. We note that, since we only focus on user-side fairness in this paper, we only apply SKIPHOP to generate embeddings of user nodes. We still use RGCN embeddings for item nodes.

For each user node $n_v \in \mathcal{V}$, we randomly sample the second-order neighbors of n_v in $\mathcal{G}$, and connect these nodes with the node n_v directly, generating a second-order graph $\mathcal{G}^2$. It is worth noting that SkipHop can be easily generalized to the neighborhood of a higher order by generating the knowledge graph $\mathcal{G}^k$ with $k > 2$.

In this paper, we consider graph attention network (GAT) [20] to capture edge-wise contributions when learning node embeddings. Then the propagation model of SkipHop that calculates the forward-pass update is formulated as below:

$$\tilde{\mathbf{h}}_u^{(l+1)} = \sigma\Big(\sum_{i \in \tilde{N}_u} \alpha_{u,i} \mathbf{W} \sigma\big(\sum_{r \in \mathcal{R}} \sum_{p \in N_i^r} \frac{\mathbf{W}_r^{(l-1)} \mathbf{h}_p^{(l-1)}}{|N_i^r|} \big) \Big), \quad (5)$$

where $\tilde{N}_u$ denotes the second-order neighbors of node u, N_i^r is the set of first-order neighbors of node i for relation r, $\mathbf{W}$ and $\mathbf{W}_r$ are training parameters, and $\alpha_{u,i}$ is the attention weight that shows the importance of the node u for node i. Specifically, we apply two layers of RGCN to obtain $\mathbf{h}_p$ and pass it to Eq. 5 to calculate $\tilde{\mathbf{h}}_u$.

B. Fairness-aware Recommendations

After the embeddings of user nodes are calculated, a straightforward approach is to generate user-item preference scores from user and item embeddings. In this paper, we use inner product as the scoring function to compute the preference scores. Specifically, the preference score of user u to item p is computed as $\hat{y}_{u,p} = \tilde{\mathbf{h}}_u^\top \mathbf{h}_p$, where $\tilde{\mathbf{h}}_u$ and $\mathbf{h}_p$ are the embeddings of u and p respectively. We follow CKE [21] and KGAT [7] and utilize Bayesian Personalized Ranking (BPR) [22] for the loss function of recommendation.

Fairness regularization (FR). Since the recommendations are not available during the training process, we design an approximation of unfairness regularization by randomly sampling k items that each user has interacted with. We use u_S to denote the k items sampled for the given user u. Then we define the *fairness regularization* as follows:

$$\mathcal{L}_{\text{FR}} = \frac{1}{|\mathcal{U}|^2} \sum_{\forall u,v \in \mathcal{U}} \mathbf{S}_{u,v} \times \Big(\frac{1}{k^2} \sum_{\forall p \in u_S} \sum_{\forall q \in v_S} d(I_p^u, I_q^v) \Big), \quad (6)$$

where I_p^u is a sampled item that user u has interacted with. $S(\cdot)$ and $d(\cdot)$ functions follow Eqs. 2 and 3 respectively. We set $\mathbf{S}_{u,v} = 0$ when $\mathbf{S}_{u,v} < \tau$ as Eq. 4 given that we only focus on similar users [13]. By adding the unfairness regularization, we have the following loss function to minimize: $\mathcal{L} = \mathcal{L}_{\text{BPR}} + \beta \mathcal{L}_{\text{FR}}$, where β is the hyperparameter that controls the trade-off between fairness and recommendation accuracy.

V. EXPERIMENTS

A. Experimental Setup

All algorithms are implemented in Python and executed on NVIDIA TITAN V GPU with 12 GB memory. We will release source code upon publication. **Datasets.** We use two datasets, namely *Movie* dataset and *Music* dataset, that are

TABLE I: Performance evaluation on Movie/Music datasets. Precision@20 (P@20), recall@20 (R@20), and NDCG@20 (N@20) are reported. UF is the unfairness score with $\gamma = 0.75$ (i.e., UF@0.75). Trade-off (TO) between fairness and accuracy is included. The best value for each metric (with or without fairness regularization) is highlighted in bold.

	P@20 (%)	R@20 (%)	N@20 (%)	UF@0.75	TO (%)
	Without fairness regularization				
NFM [23]	26.6/23.5	21.8/**10.5**	34.2/**26.9**	12.74/10.16	3.7/3.7
CKE [21]	27.1/22.4	**22.4**/10.1	**34.9**/25.4	13.14/11.24	3.9/4.5
KGAT [7]	20.9/13.5	16.7/5.7	25.4/14.3	13.06/11.81	5.2/8.4
SkipHop	**27.4**/22.6	22.0/10.3	34.6/25.0	**11.00/9.72**	**3.1/3.7**
	With fairness regularization				
NFM [23]	26.1/**23.9**	22.1/**10.7**	33.6/**27.4**	13.20/10.40	3.9/3.9
CKE [21]	27.0/21.9	**22.6**/9.9	**34.7**/25.0	12.29/10.32	3.5/3.9
KGAT [7]	19.0/11.6	15.2/4.7	22.5/12.0	12.75/11.71	5.5/9.8
SkipHop	**27.3**/22.5	22.0/10.3	34.6/24.7	**7.34/7.53**	**2.1/2.9**

popularly used in the literature [4], [5]. Movie dataset contains 6,040 users, 3,706 movies with 1 million interactions collected from MovieLens-1M. Its knowledge graph contains genre and director information of movies. We use a benchmark music dataset named *KKbox*. This music dataset contains 4,000 users, 43,268 music items, and 1,735,148 interactions. Its knowledge graph contains genre and composer information of music. **Metrics.** We adopt three commonly used metrics for evaluation accuracy of recommendations: *Precision*@K, *Recall*@K, and *NDCG*@K. For fairness evaluation, we use *Unfairness Score (UF)* (Eq. 4) as the evaluation metric. We use UF@γ to denote the value of UF under particular γ value (Eq. 2). **Baselines.** We consider the following algorithms as baselines: (1) Neural Factorization Machine (NFM) [23] that considers second-order feature interactions; (2) CKE [21] that considers collaborative knowledge base embedding; (3) KGAT [7] that uses knowledge graph attention networks. We make NFM, CKE, and KGAT fairness-aware by equipping the same fairness regularization of SkipHop with the loss function of these methods. **Parameter setting.** For both datasets, we adopt the same idea as previous work [7] and randomly choose 70% items for each user to be included in the training set, 10% items for validation set, and the remaining 20% items for the test set. We optimize all the models with Adam optimizer with batch size 256, learning rate 0.001, and weight decay $1e-5$. We fix the embedding vector dimension as 64. We choose $\beta = 0.75$ in the loss function and $\gamma = 0.75$ in Eq. 2 by a grid search in $\beta = \{1, 0.75, 0.5, 0.25\}$ and $\gamma = \{1, 0.75, 0.5, 0.25\}$ on the validation set.

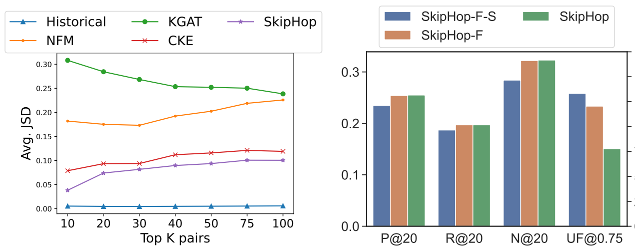
B. *Performance of* SkipHop

We evaluate the quality of the embeddings generated by SkipHop, and compare with NFM, CKE and KGAT in both accuracy (Precision@k, recall@k, NDCG@k) and the unfairness score of recommendations. The results are shown in Table I (top four rows). We denote SkipHop that does not add the fairness regularization as SkipHop-F. We have the following observations. First, on both datasets, SkipHop-F has the lowest unfairness score as well as the best trade-off between fairness and accuracy among the four methods. Indeed, SkipHop-F can achieve as large as 12% reduction in the unfairness score. This demonstrates that by incorporating the second-order proximity, SkipHop-F enhances individual fairness effectively. Second, SkipHop-F achieves competitive performance compared with the baseline methods. In particular, SkipHop-F always outperforms CKE and KGAT on both datasets. On the other hand, the recommendation accuracy of SkipHop-F is comparable with NFM on the Music dataset and outperforms NFM on the Movie dataset. Note that the unfairness score of SkipHop-F is much smaller than NFM.

Trade-off between Fairness and Accuracy We report precision@20, recall@20, NDCG@20, and the unfairness score (UG) in Table I (bottom four rows). The results show that the fairness regularization can effectively improve fairness of CKE and KGAT; their unfairness scores are improved by 6.5% and 2.3% respectively. However, the fairness regularization fails to improve the fairness of NFM. On the other hand, the accuracy of those approaches decreases. On the contrary, the accuracy of SkipHop maintains stable, while the unfairness score is further improved significantly (33.3% decrease on the Movie dataset and 22.5% decrease on the Music dataset). This makes the accuracy of SkipHop closer to the best performance of CKE, NFM, and KGAT, but with improved fairness. Next, we measure the trade-off between fairness and recommendation accuracy. We consider NDCG@20 as the accuracy metric and define the following metric to measure the trade-off (TO): $TO = \frac{UF}{NDCG@20}$. Intuitively, smaller TO values indicate better trade-off between fairness and accuracy. To make the values of NDCG@20 and UF comparable, we first project UF values into the range of [0, 1] by dividing them by $1,000$. The accuracy metric can be changed to other metrics such as precision@20 and recall@20. We show the trade-off results in Table I ("TO" column). We observe that SkipHop always has the best trade-off among all the methods. This demonstrates the effectiveness of SkipHop.

User Interests Captured by SkipHop To evaluate if SkipHop captures latent user interests, we study the genre distribution of recommended items by SkipHop and the baselines. Since individual fairness is concerned with similar users, we measure the similarity of genre distributions of recommended items for similar user pairs. We consider top-k ($k = \{10, 20, 30, 40, 50, 75, 100\}$) most similar user pairs, where user similarity is measured as Jaccard similarity of historical interactions. For each user pair <u, v>, we utilize Jensen-Shannon divergence (JSD) to measure the genre distribution distance of their recommendations R_u and R_v. Finally, we calculate the average of JSD for all k similar user pairs.

Figure 2a presents the result of average JSD distance of genre distribution by SkipHop, historical items, and three baseline methods on Movie dataset. It shows that the items recommended by SkipHop have more similar genre distribution than three baselines. Specifically, the average JSD by SkipHop never exceeds 0.1, even when as many as top-100 similar user pairs are considered. This demonstrates that the recommendations by SkipHop better capture the true



(a) (b)

Fig. 2: (a): Comparison of genre distributions of items recommended by SKIPHOP and baselines for the top-k similar pairs of users on the Movie dataset. (b): Ablation study for three settings: SKIPHOP, SKIPHOP-F, and SKIPHOP-F-S on the Movie dataset. Left y-axis is for P@20, R@20, N@20; Right y-axis is for UF@0.75.

distribution of users' historical interactions than the baselines.

Ablation Study We consider three settings: SKIPHOP, SKIPHOP without fairness regularization (SKIPHOP-F), and SKIPHOP without both fairness constraint and second-order connections (SKIPHOP-F-S). The results on the Movie dataset are shown in Figure 2b. First, SKIPHOP and SKIPHOP-F outperform SKIPHOP-F-S in terms of both accuracy and fairness. It demonstrates that the second-order connections are useful to achieve better accuracy performance. Second, without the fairness regularization on SKIPHOP, the unfairness score of recommendations increased, which indicates the effectiveness of fairness regularization in bias mitigation. We conclude that SKIPHOP is effective to address the trade-off between fairness and recommendation accuracy.

VI. CONCLUSION AND DISCUSSION

In this work, we study the problem of enforcing recommender systems on knowledge graphs with individual user fairness. First, we formally define *individual fairness*, which requires that similar users should receive similar recommendations. As part of this fairness notion, we define the similarity metrics that incorporate both first-order and second-order proximity. Second, we design a novel graph neural network named SKIPHOP that models the users' latent interests from their second-order proximity. Our experimental results demonstrate the effectiveness of SKIPHOP in terms of fairness and recommendation accuracy compared with the state-of-the-art recommender systems. This work can be extended in several exciting directions for future work. We will investigate how to extend SKIPHOP to group fairness notions. We will study the impact of integrating higher-order proximity with SKIPHOP in terms of both fairness and recommendation accuracy.

ACKNOWLEDGMENTS

This work is supported in part by the US National Science Foundation under grants 1948432, 2047843, 2029038, and 2135988.

REFERENCES

[1] B. Rastegarpanah, K. P. Gummadi, and M. Crovella, "Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems," in *ACM WSDM*, 2019, pp. 231–239.
[2] M. D. Ekstrand, R. Burke, and F. Diaz, "Fairness and discrimination in retrieval and recommendation," in *SIGIR*, 2019, pp. 1403–1404.
[3] S. Yao and B. Huang, "Beyond parity: Fairness objectives for collaborative filtering," in *NeurIPS*, 2017, pp. 2921–2930.
[4] L. Xiao, Z. Min, Z. Yongfeng, G. Zhaoquan, L. Yiqun, and M. Shaoping, "Fairness-aware group recommendation with pareto-efficiency," in *ACM RecSys*, 2017, pp. 107–115.
[5] Z. Fu, Y. Xian, R. Gao, J. Zhao, Q. Huang, Y. Ge, S. Xu, S. Geng, C. Shah, Y. Zhang *et al.*, "Fairness-aware explainable recommendation over knowledge graphs," in *SIGIR*, 2020, pp. 69–78.
[6] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012, p. 214–226.
[7] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *KDD*, 2019, pp. 950–958.
[8] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *ICML*, 2019, pp. 21–29.
[9] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *KDD*, 2015.
[10] T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," *Data Mining and Knowledge Discovery*, vol. 21, no. 2, pp. 277–292, 2010.
[11] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *NeurIPS*, 2016.
[12] M. Kim, O. Reingold, and G. Rothblum, "Fairness through computationally-bounded awareness," *NeurIPS*, vol. 31, pp. 4842–4852, 2018.
[13] J. Kang, J. He, R. Maciejewski, and H. Tong, "Inform: Individual fairness on graph mining," in *KDD*, 2020, pp. 379–389.
[14] Y. Li, H. Chen, Z. Fu, Y. Ge, and Y. Zhang, "User-oriented fairness in recommendation," in *The Web Conf*, 2021, pp. 624–632.
[15] A. J. Biega, K. P. Gummadi, and G. Weikum, "Equity of attention: Amortizing individual fairness in rankings," in *SIGIR*, 2018, pp. 405–414.
[16] V. Do, S. Corbett-Davies, J. Atif, and N. Usunier, "Online certification of preference-based fairness for personalized recommender systems," in *AAAI*, vol. 36, no. 6, 2022, pp. 6532–6540.
[17] Y. Li, H. Chen, S. Xu, Y. Ge, and Y. Zhang, "Towards personalized fairness based on causal notion," in *SIGIR*, 2021, pp. 1054–1063.
[18] G. K. Patro, A. Biswas, N. Ganguly, K. P. Gummadi, and A. Chakraborty, "Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms," in *The Web Conf*, 2020, pp. 1194–1204.
[19] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.
[20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
[21] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *KDD*, 2016, p. 353–362.
[22] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, p. 452–461.
[23] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *SIGIR*, 2017.