

# Mitigation of Popularity Bias in Recommendation Systems

Sabrina. Karboua<sup>1,3,\*</sup>, Fouzi. Harrag<sup>1,3</sup>, Farid. Meziane<sup>2</sup> and Amal. Boutadjine<sup>1,3</sup>

<sup>1</sup>Departement of Computer Science, College of Sciences, Ferhat ABBAS Setif 1 University, 19000 Setif, Algeria

<sup>2</sup>Data Science Research Centre, University of Derby Markeaton Street, Derby DE22 3AW, UK

<sup>3</sup>Mechatronics Laboratory, Optics and Precision Mechanics Institute, Ferhat ABBAS University Setif 1, 19000 Setif, Algeria

## Abstract

In response to the quantity of information available on the Internet, many online service providers are attempting to customize their services and make content access more simple via recommender systems (RSs) to support users in discovering the products they are most likely interested in. However, these recommendation systems are prone to popularity bias, which is a tendency to promote popular items even if they do not satisfy a user's preferences and then provide customers with recommendations of poor quality. Such a bias has a negative influence on both users and item providers. It is then essential to mitigate such bias in order to guarantee that less popular but pertinent items show up on the user's recommendation list. In this work, we conduct an empirical analysis of different mitigation techniques for popularity bias to provide an overview of the present state of the art of popularity bias and raise the fairness issue in RSs.

## Keywords

Popularity bias, Recommender System, Fairness, Mitigation

## 1. Introduction

Recommendation systems (RSs) have been enormously effective in solving the issue of information overload, in which users struggle to discover information they are interested in. We encounter RSs, for example, on Netflix to recommend movies, on Spotify for music recommendations, on online educational websites to recommend courses like Coursera, when we get purchase recommendations on e-commerce websites like Amazon, and when we are advised of new connections on social networking platforms like Facebook. A recommender system's performance is often assessed in terms of multiple criteria like accuracy, diversity, novelty, and fairness. Fairness in RSs is a notion that has lately gained a lot of interest because developing biased recommendation systems inhibits customers from discovering items that are not extremely popular yet are good fits for them. Popularity bias is one factor that contributes to unfairness in RSs as it prevents diverse items from having an equal opportunity of recommendation and exposure. This is a phenomenon in which the most popular items get more

---

Tunisian Algerian Conference on Applied Computing (TACC 2022), December 13–14, 2022, Constantine, Algeria

✉ sabrina.karboua@univ-setif.dz (Sabrina. Karboua); fouzi.harrag@univ-setif.dz (Fouzi. Harrag);

F.Meziane@derby.ac.uk (Farid. Meziane); boutadjine.amal@univ-setif.dz (Amal. Boutadjine)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

and more exposure, while less popular items receive less exposure [1]. Popularity bias occurs because popular items often have significantly more rating data than less popular ones. This is because the recommender systems are trained on user preferences, and normally, many users rate the popular items while the less popular items earn just a few ratings, which implies that the unpopular items that constitute the so-called long tail of recommendations, do not gain enough exposure, particularly when they are new to the system [2]. Training RSs models on biased data promotes the recommendation of popular items more often than less popular ones. The more rating data the items have, the more they are suggested, and the more the items get recommended, the more ratings they earn from users, and then "the rich get richer, and the poor get poorer" [3].

## 2. Definition of Popularity Bias

Various researchers have studied the idea of popularity bias, which is also known by other names like Matthew effect [4], long-tail recommendation [5, 6] and aggregate diversity [7]. All of these words refer to the fact that just a few popular items are regularly recommended, whereas most unpopular items are seldom recommended. The popularity bias of item  $i$  is calculated mathematically by dividing the number of users who gave item  $i$  a rating by the total number of users [1].

$$POP(i) = \frac{\sum_{u \in U} 1(i \in p_u)}{|U|},$$

where  $U$  represents the list of users and  $p_u$  represents the list of items rated by user  $u$ .

## 3. Empirical Examples of Popularity Bias and Its Impacts in RSs

Popularity bias has a negative influence not only on the consumers of a recommender system, but also on the providers, and the system itself. Bias toward popular items may influence consumption of less popular ones, hinders them from being popular in the future and then harm the recommendation fairness. The effects of this bias can be better understood through real-world examples. According to Abdollahpouri et al. [3], popularity bias homogenizes the market by allowing only a few item producers to dominate it, resulting in fewer chances for innovation and originality.

Boratto et al. [8] state that in the educational field, an online educational platform recommendation may exhibit undesired bias and subsequent educational amplifications. They demonstrate that a few popular courses evaluated by a large number of learners might cause the platform to suffer from popularity bias, preventing new courses from appearing at the top of the suggested list. As a result, the platform may be controlled by a few well-known courses and consequently few teachers.

In the music and movie domains, the analysis by [3, 9, 10] reveals that most of items in the long-tail category do not receive enough exposure and attention, whereas few popular items

are recommended regularly. They highlight also that majority of consumers that have little interest in popular songs/ movies, would be disadvantaged by the recommendation algorithm since it tends to recommend popular items, and this leads to an under-recommendation of unpopular items. Consequently, users that are interested in unpopular songs/ movie will get lower recommendation than those interested in popular music/ movie.

The findings of [11, 12] in the book domain show that the most frequently used algorithms are unable to pique users' interest in niche books and instead promote largely popular books, and users with niche preferences get much poorer quality recommendation than users with mainstream tastes.

The research [13] investigates the presence of popularity bias in the online dating websites and its implications for users' chances of meeting dating partners. They find that even if the recommendation algorithm recommends popular users more often, popular users are likely to accept invitations and messages from those other non-popular users, leading to inferior matching results.

Gharahighehi al.[14] state that it is quite likely that RSs promote the popularity bias by recommending only very popular articles, ignoring articles by less popular authors whether their writings are important to a certain set of readers. This unfair recommendation might have detrimental effects for these journalists, as they may eventually lose faith in the platform, affecting other parties as well as the whole system.

## 4. Mitigation Techniques

Most popularity bias mitigation algorithms are categorized into two types: in-processing algorithms and post-processing algorithms.

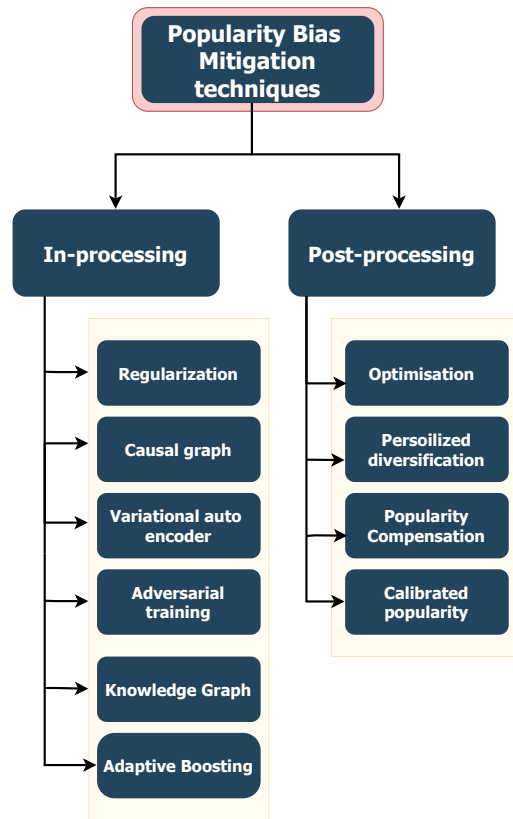
### 4.1. In-processing Techniques

These approaches include altering the recommender system algorithm during training or introducing new RSs algorithms to incorporate fairness into the model.

**1- Causal Graph.** A causal graph is a directed acyclic graph, in which each node represents a variable and each edge represents a causal relationship between two nodes. Zhang et al. [15] demonstrate that between the exposed items and the observed interactions, item popularity functions as a confounder, resulting in the negative effect of bias amplification. They use the causal graph to analyze how item popularity influences the recommendation process.

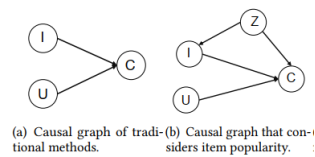
The causal graph of traditional recommender methods only considers that the interaction probability ( $C$ ) of an exposed item ( $I$ ) that is consumed by user  $U$  is determined by the match between user interest and item property ( $U, I \rightarrow C$ ). Zhang et al. [15] enrich the graph

by adding the item popularity represented as node Z, which considers that the interaction probability is based on the user, the exposed item, and the popularity of item ( $U, I, E \rightarrow C$ ).



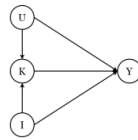
**Figure 1:** Classification of popularity bias mitigation techniques

Then, they conduct a deconfounded training with do-calculus and back-door criteria to eliminate the negative influence of the confounder and intervened the popularity bias during the recommendation inference that provides top-K recommendation list via causal intervention.



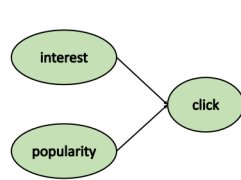
**Figure 2:** proposed causal graph [15]

To overcome popularity bias, Wei et al. [16] disentangle the effects of item popularity and user interest from causal inference. They first create a causal graph to represent the key cause-effect relationships in the recommendation model, and assume that the probability of an interaction is influenced by the user-item matching, user conformity and item popularity. Then, they propose a model-agnostic counterfactual reasoning (MACR) framework for training the causal graph-based recommender model and perform counterfactual inference in the recommendation inference step to reduce popularity bias.



**Figure 3:** proposed causal graph [16]

Another study similar to [16] is produced by Zheng et al. [17]. They make the assumption that each click is the result of union of two separate causes: item's popularity and user's interest. Then, they employ different embeddings for the user's interest and item's popularity, in order to have two embeddings for each user or item. After that, they train the model under the framework of multi-task learning using cause-specific data to constrain each embedding to represent only one cause and then achieve disentanglement between popularity embedding and interest embedding. Furthermore, direct disentanglement supervision is used to achieve greater independence across embeddings of various causes.



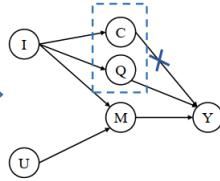
**Figure 4:** proposed causal graph by [17]

Wang et al. [18] present a novel framework DecRS that explicitly incorporates causal effect of user representation on prediction score and used an approximation operator for backdoor adjustment to eliminate the misleading correlation induced by the confounder, and then present a user-specific inference technique for dynamically regulating the impact of backdoor adjustment based on user state.

He et al. [19] present a novel framework named Mitigating Popularity Prejudice in Recommendation through Counterfactual Inference (MPCI) to eliminate popularity bias from both the data and model perspectives. MPCI used click data to capture user preference representations

and then extracted biased data to learn popularity bias representations to estimate how popularity bias affects the prediction score in a causal graph, which may then be used to mitigate its detrimental effect by counterfactual inference. A collection of both positive and negative samples are included in each representation in order to measure the popularity bias prediction score and the user preference prediction score using a fusion function. The two scores and the expectation constant are then combined, and used as the counterfactual inference’s input to finally predict the user’s prediction score for each item.

Zhao et al. [19] state that not all popularity bias is harmful, because some items are more popular because they have intrinsically better properties, and removing the popularity bias completely would deteriorate the model performance. They argue that popularity bias is affected by two factors: static item quality and the dynamic conformity effect (which refers to user who follows community standards while diverging from his personal preferences). Therefore, it is critical to distinguish between the desirable popularity bias induced by item quality and conformity that leads to bad popularity bias.



**Figure 5:** proposed causal graph by [19]

To achieve this goal, they propose a novel Time-aware DisEntangled framework (TIDE) using causal graph to model that a click is generated by three component: conformity effect ( $C$ ), item’s quality ( $Q$ ) and item-user matching score( $M$ ) returned by the recommendation model. Then, they apply a causal intervention during the inference stage on the conformity module to help the prediction avoids the undesirable conformity impact and benefits from the item quality and interest matching.

**2- Variational Auto Encoder.** VAEs are Multi Layer Perceptrons (MLP)-based generative models that have recently been presented as a strong method for building Collaborative Filtering (CF) recommenders and minimizing bias.

To reduce popularity bias and boosting diversity in results, Borges et al. [20] propose to adapt the VAE by adding a penalty constraint to the reconstruction error (decoder) to boost the visibility of unpopular items as follow:

$$\log p(x_u|z_u) = \sum_{i=0}^N x_{ui} \log \pi_i(z_u) \cdot \lambda,$$

The penalty term  $\lambda$  has the effect of lowering the score of the most popular of all items by multiplying it by 0, while keeping the niche items at their initial score multiplied by 1, and it is defined as follow:

$$\lambda = 1 - \frac{w(x_i) - \min(w(x))}{\max(w(x)) - \min(w(x))},$$

where  $w(x)$  returns numbers of interactions for item  $x$ .

**3- Regularization.** It has been used in matrix factorisation algorithms to control overfitting by limiting the size of the latent components.

Kiswanto et al. [21] propose a fairness-aware regularization learning-to-rank method to make the weight of unpopular items higher without affecting the recommender system ranking performance by recommending a balanced mix of popular and unpopular items to users.

Abdollahpouri et al. [22] investigate how regularization may be used to mitigate the recommender system's popularity bias. They start with an optimization goal of the form:

$$\min_{P,Q} acc(P, Q) + \lambda reg(P, Q),$$

where  $acc$  represents the accuracy target,  $reg$  is the regularization term, in which the  $LapDQ$  regularizer was used, and  $\lambda$  is a coefficient for managing the regularizer effect. The  $LapDQ$  regularizer has the form  $tr(Q^T L_D Q)$ , where  $tr$  is the trace of the matrix  $D$ ,  $L_D$  is the Laplacian matrix of  $D$ .  $D_{i,j} = 1$  if item  $i$  and  $j$  are from the same set (popular or unpopular items), and 0 if items  $i$  and  $j$  are from different set, and  $Q$  is the item latent representation matrix.

Zhu et al. [23] calculate the correlation between the predicted preference score for a positive user-item and the popularity of matched items using the square of the Pearson regularizer and then, reduce the bias by reducing this regularization term coupled with the recommendation error using the following form:

$$\min_{P,Q} L_{Rec} + \lambda PPC(\hat{R}_+, pop(I))^2,$$

where  $L_{Rec}$  represents the loss of recommendation models,  $\lambda$  is the trade-off weight, and  $PPC(\cdot)$  is the Pearson correlation coefficient between expected scores and items popularity.  $\hat{R}$  is the predicted user-item preference matrix, and  $POP(\cdot)$  is the item popularity.

**4- Adversarial Training.** Krishnan et al.[24] propose an adversarial training technique to mitigate the bias. The adversary network  $D$  learns in the feedback data the implicit relationship structure of items, and correlate niche item recommendation of the base recommender  $G$  with popular items in the user's history, while the base recommender model  $D$  is concurrently trained with the adversary network to replicate these associations while avoiding the adversarial penalty until mutual convergence. More specifically, The adversary  $D$  is taught to differentiate synthetic pairs of popular and unpopular items selected from  $G$  and real pairings of popular

and unpopular items obtained from the global co-occurrence of matrix  $X$ . When the synthetic and real niche-popular pairings match with the association structure acquired by  $D$ , the model converges.

**5- Knowledge Graph.** Based on the essential assumption that a user often has several preferences prompting him to consume diverse items, Wei et al. [25] suggest a framework to mitigate popularity bias from the users' perspective using knowledge graph, which profiles user-item connections throughout the knowledge graph using fine-grained preferences, and then eliminate a proportion of popularity preference for distinct users. The framework seeks to apply Knowledge Graph integrated with popularity nodes to decrease popularity bias by determining the fine-grained preferences of users. It initially builds a heterogeneous network by combining Knowledge Graph, preference graph, and popularity nodes. The embeddings of item, user, preference and the mutual attention parameters were then learned by applying a heterogeneous graph transformer to the heterogeneous graph while matching fine-grained preferences with the relations in Knowledge Graph. Finally, popularity preference is removed adaptively based on the user's interest in popular items to reduce the bias.

**6- Adaptive Boosting.** Inspired by the fair boosting technique on classification. Gangwar et al. [26] provide an algorithm named "FairBoost" that minimizes the popularity bias existing in the data while preserving accuracy within reasonable bounds, by increasing the weights of the unpopular items, which are typically under-represented in the data and then, maintain a balance between popular and unpopular items.

## 4.2. Post-Processing Techniques

This kind of techniques may be applied to any recommender system's output by reordering the original recommended list based on some constraints.

**1- Knapsack Optimization model.** Seymen et al.[27] propose a re-ranking optimization model to find the best system solution given diverse constraints to deal with various different problems like popularity bias, diversity and provider fairness. They start by defining a baseline model to increase the overall recommended items' average predicted ratings, with the restriction that each user gets  $K$  recommendation.

$$\frac{1}{K|U|} \sum_{i \in I} \sum_{u \in U} r_{iu} x_{iu},$$

where  $I$  is the set of items and  $U$  is the set of users,  $K$  is the number of items to be recommended to each user,  $r_{iu}$  represents the predicted rating for user  $u \in U$  and item  $i \in I$ .  $x_{ij}$  is a binary value (an item is either recommended or not) that specifies items that are recommended, where  $x_{ij} = 1$  if item  $i$  is recommended to user  $u$  and 0 otherwise.

Then, They extend the base model by adding a knapsack constraint (Pop-Opt) to leverage the popularity of all recommended items as follows.

$$\sum_{i \in I} \sum_{u \in U} x_{iu} \omega_i \leq \alpha,$$



where  $\alpha$  represents the maximum bound on the overall popularity of the recommended items and  $\omega_i$  calculates the item's popularity  $i$  by dividing the number of ratings received by item  $i$  by the sum of all the ratings for all other items in the system.

**2- Personalized diversification approach.** To decrease the effect of popularity bias, a new post-processing method is proposed by Abdollahpouri et al. [28] by reordering the initial recommendation list provided by the recommender algorithm. Their 'diversification technique' aim is to achieve the required compromise between accuracy and long-tail coverage (improve the representation of underrepresented items). Their strategy is based on an algorithm named the eXplicit Query Aspect Diversification (xQuAD), which is developed to diversify query results so it addresses a wide range of query components. They apply the following criterion on the final recommendation list to maintain a balance between ratio of popular items and niche items as follow:

$$P(v|u) + \lambda P(v, S'|u),$$

$P(v|u)$  indicates the probability that a user  $u \in U$  (list of users) is interested in item  $v \in V$  (list of items), and  $P(v, S'|k)$  reflects the probability that user  $u$  will be interested in item  $v$  and that  $v$  is not already in  $S$  (new re-ordered list). The first term emphasizes diversity between popular and unpopular items, whereas the second term encourages scoring accuracy. In general,  $\lambda$  parameter affects how highly the regulating of popularity bias is weighted. The item with the highest score is included in the final list  $S$ , then repeated the procedure until  $S$  achieves required length. In order to produce more diversified recommendation that includes both popular  $\Gamma$  and unpopular  $\Gamma'$  items, the marginal probability  $P(v, S'|u)$  is computed as:

$$P(v, S'|u) = \sum_{d \in \Gamma, \Gamma'} P(d|u) P(v|d) \prod_{i \in S} (1 - P(i|dS)),$$

where  $P(d|u)$  is an indicator of user preference across distinct item groups,  $P(d|v)$  is equal to 1 if item  $i$  in the original recommended list  $S$  includes category  $d$  and 0 otherwise.

**3- Calibrated popularity.** A distributional discrepancies in the groups (short-head, medium, tail) to which items belong between the user's profile and its recommended list is measured. The principle of calibrated popularity (CP) is that for example, if a user likes 10% popular items, 30% items of medium popularity, and 40% unpopular items, the recommended list should comprise 10% popular items, 30% items of medium popularity, and 40% niche items [29]. The Calibrated Popularity method generates a final recommendation list  $L_u$  for each user  $u$  from an initial recommended list  $S$  created by a base recommender. A weighted sum of relevance and calibration is used and maximized as follow

$$F_u = \arg \max (1 - \lambda) \cdot Rel(L_u) - \lambda \cdot \mathfrak{S}(P, Q(L_u)),$$

where  $\lambda$  is the weight regulating popularity calibration vs the relevance,  $Rel(L_u)$  is the total of the predicted scores for items in  $L_u$ .  $\mathfrak{S}$  represents the jenson-shanon divergence between the recommendation profile and recommendation list items.

**4- Popularity Compensation.** Another research [23] introduces another re-ranking strategy that alters the predicted user-item preference matrix by compensating low-popularity items so that they have greater preference scores and therefore, they are ranked higher. This compensation is based on three key principles:

- Should be dependent on item popularity, with less popular items receiving higher compensation.

- Compensation should be also dependent on user preferences; items that have higher possibility of being preferred by a user should be rewarded more. The algorithm therefore assures that items that are not liked by a user and has a low popularity will not be recommended by mistake.

- Compensation should be based on each user's value scale: item candidates for a user with a higher value scale should be compensated more. This ensures that consumers with high value scales of estimated scores receive adequate compensation for items, ensuring that the algorithm is useful to all users.

For one item  $i$  given user  $u$ , the popularity compensation score is calculated using the following equation:

$$C_{u,i} = \frac{1}{pop(i)} \cdot (\hat{R}_{u,i} \cdot \beta + 1 - \beta),$$

where  $\hat{R}_{u,i}$  is the predicted preference scores from user  $u$  to items produced by the algorithm.  $1/pop(i)$  used to achieve the first condition.  $\hat{R}_{u,i} \cdot \beta + 1 - 1$  to achieve the second condition.  $\beta \in [0, 1]$  is a trade-off weight used to adjust the predicted preference score ratio in the compensation.

Condition 3 is calculated using the following equation:

$$\hat{R}_{u,i}^* = \hat{R}_{u,i} \cdot \alpha + C_{u,i} \cdot n_u / m_u,$$

where  $\hat{R}_{u,i}^*$  is the new preference score from  $u$  to  $i$ ,  $\alpha$  represents the algorithm's trade-off weight, and  $n_u/m_u$  used to normalize the compensation scores ( $n_u$  represents predicted scores norm for user  $u$  and  $m_u$  is the compensation scores norm of  $u$  excepting those for items in the interacted item set in the training data).

## 5. Conclusion

In this paper, we explained the issue of popularity bias in Recommenders System (RSs), then we systematically provided a mitigation techniques classification into two approaches in-processing (model based) and post-processing (re- ranking). The first approach is based on improving current algorithms by adding a constraint into the objective function or propose new algorithms. In the second approach, the original recommended list is re-ordered based on some constraints to mitigate the popularity bias.

## References

- [1] H. Abdollahpouri, Popularity bias in recommendation: a multi-stakeholder perspective, Ph.D. thesis, University of Colorado at Boulder, 2020.

- [2] E. Yalcin, A. Bilge, Investigating and counteracting popularity bias in group recommendations, *Information Processing & Management* 58 (2021) 102608.
- [3] H. Abdollahpouri, M. Mansoury, R. Burke, B. Mobasher, The unfairness of popularity bias in recommendation, *arXiv preprint arXiv:1907.13286* (2019).
- [4] J. Möller, D. Trilling, N. Helberger, B. van Es, Do not blame it on the algorithm: an empirical assessment of multiple recommender systems and their impact on content diversity, *Information, Communication & Society* 21 (2018) 959–977.
- [5] H. Abdollahpouri, R. Burke, B. Mobasher, Controlling popularity bias in learning-to-rank recommendation, in: *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 42–46.
- [6] H. Yin, B. Cui, J. Li, J. Yao, C. Chen, Challenging the long tail recommendation, *arXiv preprint arXiv:1205.6700* (2012).
- [7] G. Adomavicius, Y. Kwon, Improving aggregate recommendation diversity using ranking-based techniques, *IEEE Transactions on Knowledge and Data Engineering* 24 (2011) 896–911.
- [8] L. Boratto, G. Fenu, M. Marras, The effect of algorithmic bias on recommender systems for massive open online courses, in: *European Conference on Information Retrieval*, Springer, 2019, pp. 457–472.
- [9] D. Kowald, M. Schedl, E. Lex, The unfairness of popularity bias in music recommendation: A reproducibility study, in: *European conference on information retrieval*, Springer, 2020, pp. 35–42.
- [10] C. Bauer, M. Schedl, Global and country-specific mainstreamness measures: Definitions, analysis, and usage for improving personalized music recommendation systems, *PloS one* 14 (2019) e0217389.
- [11] M. Naghiaei, H. A. Rahmani, M. Dehghan, The unfairness of popularity bias in book recommendation, *arXiv preprint arXiv:2202.13446* (2022).
- [12] M. Naghiaei, H. A. Rahmani, M. Dehghan, The unfairness of popularity bias in book recommendation, *arXiv preprint arXiv:2202.13446* (2022).
- [13] M. Celdir, S.-H. Cho, E. H. Hwang, Popularity bias in online dating platforms: Theory and empirical evidence, Available at SSRN 4053204 (2022).
- [14] A. Gharahighehi, C. Vens, K. Pliakos, Fair multi-stakeholder news recommender system with hypergraph ranking, *Information Processing & Management* 58 (2021) 102663.
- [15] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, Y. Zhang, Causal intervention for leveraging popularity bias in recommendation, in: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 11–20.
- [16] T. Wei, F. Feng, J. Chen, Z. Wu, J. Yi, X. He, Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system, in: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1791–1800.
- [17] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, D. Jin, Disentangling user interest and conformity for recommendation with causal embedding, in: *Proceedings of the Web Conference 2021*, 2021, pp. 2980–2991.
- [18] W. Wang, F. Feng, X. He, X. Wang, T.-S. Chua, Deconfounded recommendation for alleviating bias amplification, in: *Proceedings of the 27th ACM SIGKDD Conference on*

- Knowledge Discovery & Data Mining, 2021, pp. 1717–1725.
- [19] M. He, C. Li, X. Hu, X. Chen, J. Wang, Mitigating popularity bias in recommendation via counterfactual inference, in: International Conference on Database Systems for Advanced Applications, Springer, 2022, pp. 377–388.
  - [20] R. Borges, K. Stefanidis, On mitigating popularity bias in recommendations via variational autoencoders, in: Proceedings of the 36th Annual ACM Symposium on Applied Computing, 2021, pp. 1383–1389.
  - [21] D. Kiswanto, D. Nurjanah, R. Rismala, Fairness aware regularization on a learning-to-rank recommender system for controlling popularity bias in e-commerce domain, in: 2018 International Conference on Information Technology Systems and Innovation (ICITSI), IEEE, 2018, pp. 16–21.
  - [22] H. Abdollahpouri, R. Burke, B. Mobasher, Controlling popularity bias in learning-to-rank recommendation, in: Proceedings of the eleventh ACM conference on recommender systems, 2017, pp. 42–46.
  - [23] Z. Zhu, Y. He, X. Zhao, Y. Zhang, J. Wang, J. Caverlee, Popularity-opportunity bias in collaborative filtering, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 85–93.
  - [24] A. Krishnan, A. Sharma, A. Sankar, H. Sundaram, An adversarial approach to improve long-tail performance in neural collaborative filtering, in: Proceedings of the 27th ACM International Conference on information and knowledge management, 2018, pp. 1491–1494.
  - [25] F. Wei, S. Chen, J. Jin, S. Zhang, H. Zhou, Y. Wu, Adaptive alleviation for popularity bias in recommender systems with knowledge graph, Security and Communication Networks 2022 (2022).
  - [26] A. Gangwar, S. Jain, An adaptive boosting technique to mitigate popularity bias in recommender system, arXiv preprint arXiv:2109.05677 (2021).
  - [27] S. Seymen, H. Abdollahpouri, E. C. Malthouse, A unified optimization toolbox for solving popularity bias, fairness, and diversity in recommender systems., in: MORS@ RecSys, 2021.
  - [28] H. Abdollahpouri, R. Burke, B. Mobasher, Managing popularity bias in recommender systems with personalized re-ranking, in: The thirty-second international flairs conference, 2019.
  - [29] H. Abdollahpouri, M. Mansoury, R. Burke, B. Mobasher, E. Malthouse, User-centered evaluation of popularity bias in recommender systems, in: Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, 2021, pp. 119–129.