ReBole 2.0: Towards a More Up-to-Date Recommendation Library

Wayne Xin Zhao
Chen Yang
Jingsen Zhang
Wenqi Sun
Gaowei Zhang
Shanlei Mu
Renmin University of China
Beijing, China

Yupeng Hou†
Zeyu Zhang
Shuqing Bian
Yushuo Chen
Zhen Tian
Xinyan Fan
Ji-Rong Wen
Renmin University of China
Beijing, China

Xingyu Pan†
Zihan Lin
Jiakai Tang
Lanling Xu
Changxin Tian
Xu Chen∗
Renmin University of China
Beijing, China

ABSTRACT

In order to support the study of recent advances in recommender systems, this paper presents an extended recommendation library consisting of eight packages for up-to-date topics and architectures. First of all, from a data perspective, we consider three important topics related to data issues (i.e., sparsity, bias and distribution shift), and develop five packages accordingly, including meta-learning, data augmentation, debiasing, fairness and cross-domain recommendation. Furthermore, from a model perspective, we develop two benchmarking packages for Transformer-based and graph neural network (GNN)-based models, respectively. All the packages (consisting of 65 new models) are developed based on a popular recommendation framework ReBole, ensuring that both the implementation and interface are unified. For each package, we provide complete implementations from data loading, experimental setup, evaluation and algorithm implementation. This library provides a valuable resource to facilitate the up-to-date research in recommender systems. The project is released at the link: https://github.com/RUCAIBox/RecBole2.0.

CCS CONCEPTS

• Information systems → Personalization.

KEYWORDS

Recommendation library; Reproducibility; Evaluation

ACM Reference Format:
Wayne Xin Zhao, Chen Yang, Jingsen Zhang, Wenqi Sun, Gaowei Zhang, Shanlei Mu, Yupeng Hou†, Zeyu Zhang, Shuqing Bian, Yushuo Chen, Zhen Tian, Xinyan Fan, Ji-Rong Wen, Xingyu Pan†, Zihan Lin, Jiakai Tang, Lanling Xu, Changxin Tian, and Xu Chen∗. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM ’22), October 17–21, 2022, Atlanta, GA, USA. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557680

1 INTRODUCTION

Nowadays, recommender systems have deeply revolutionized people’s daily life, bringing a huge amount of business value and great convenience for information seeking. In the literature, various recommendation algorithms have been proposed based on different architectures or approaches [28]. Despite the great progress in recommender systems, there are increasing concerns on the reproducibility of the recommendation algorithms [1, 30]. Facing with this issue, a number of open-sourced recommendation libraries have been released to facilitate the reproducible implementation of the proposed recommendation algorithms [1–3, 5, 10, 18, 21, 23, 30, 31]. These libraries largely enhance the reproducibility of recommendation algorithms for research purpose.

However, existing recommendation libraries mainly focus on classical models, lacking consideration of the recent advances in recommender systems, including both new models (e.g., graph neural networks [25]) and new topics (e.g., debiasing [4] and fairness [19]). Due to the rapid progress in recommender systems, we argue that a more up-to-date recommendation library is needed for supporting the research on new advances. It is particularly important to standardize these ongoing studies at an early stage, preventing non-standard implementation or unreliable evaluation.

Inspired by this motivation, this paper presents a significant extension of a previously released recommendation library RecBole [30]1 (receiving extensive attention of about 2K stars on GitHub), by incorporating a series of benchmarking packages for up-to-date advances in recommender systems. In particular, our extension is conducted in two major aspects, namely data and model. First of all, there is increasing attention on the issues from the interaction data itself [4], and thus we focus on three important research topics related to data issues, namely data sparsity, data bias and data distribution shift. Considering the three data issues, we develop five benchmarking packages corresponding to meta-learning, data...
We present the overall introduction of the extended library in Figure 1. As we can see, our extensions consist of eight packages, grouped into two major parts concerning data and model. For the data packages, we focus on three key issues, that is, data sparsity, data bias and data distribution shift. To solve the data sparsity issue, we develop two packages, namely data augmentation (generating additional data samples for optimizing recommendation models) and meta-learning based recommendation (approaching the cold-start issue with meta-learning methods). To alleviate the data bias, we develop two packages, namely debiased (reducing the data bias) and fairness recommendation (enforcing the fairness of the recommendations). To overcome the data distribution shift problem, we implement a cross-domain recommendation package. Note that cross-domain recommendation is not a new research topic, while our focus is to include the most up-to-date cross-domain recommendation models. For the model packages, we aim to collect and implement the most up-to-date recommender models. Specifically, we focus on graph neural network and Transformer based models.

Besides, we also implement an application package for the task of person-job fit [6], which has received much attention from the research community. In our library, the optimal configurations (obtained by grid search in predefined parameter ranges) for all the models have been provided, based on which one can easily produce the benchmarking results for each package. In the following, we introduce the above packages in detail.

3 PACKAGE DETAILS AND USAGE

We organize the packages in two groups, namely data-oriented and model-oriented packages, which are detailed below.

3.1 Data-oriented Packages

Data augmentation (RecBole-DA). To alleviate the data sparsity problem, a recently proposed technique is to generate additional samples to densify the original user-item interactions. Following [16], we implement three types of models based on different data augmentation strategies, including heuristic methods, model based methods and hybrid methods. Besides providing model implementations, we also provide easy-to-use APIs to configure and combine different data augmentation strategies.

Meta recommendation (RecBole-MetaRec). Originated from computer vision and machine learning, meta-learning is a principled approach to dealing with few-shot learning tasks. We implement three types of meta-learning recommendation models, i.e., prediction, parameterization and embedding. These models are implemented by a series of general modules (MetaDataset, MetaDataLoader, MetaRecommender, MetaTrainer, MetaCollector and MetaUtils), which are flexible to be extended with new models.

Fairness recommendation (RecBole-FairRec). The second package related to data bias is targeted at fairness recommendation [27], considering the data bias from user perspective (debiased recommendation package mainly focuses on the bias from item side). Specifically, we implement four models in this package including FOFC [26], PFCN [14], FairGo [24] and NFCF [11]. Besides models, we implement a series of fairness metrics, which are particularly important for fairness recommendation, including Gini Index [8], Popularity Rate [8], Differential Fairness [11], Value Unfairness, Absolute Unfairness, Underestimation Unfairness, Overestimation Unfairness [26] and Non-Parity Unfairness [12].

Debiased recommendation (RecBole-Debias). Data bias ubiquitously exists in the observed user-item interaction data in recommender systems [4]. To correct these biases, we implement six debiased models considering selection bias and exposure bias. Besides, we also implement specific dataloaders for three major debiasing datasets (Yahoo!R3 [20], ML-100K [20] and KuaiRec [7]) for conveniently reproducing the experiments.
which makes Transformer play different roles in sequence modeling. Specifically, in sequential recommendation, Transformer is utilized to encode text semantics. In news recommendation, although the basic Transformer architecture has been shown effective to model topic information, there are three steps: (1) indicate the configuration file, and (2) launch the program by indicating the model name and dataset. An example of running PDA [29] based on the quick-start script. An example to train and evaluate FOCF [26] is shown in Figure 2(c). In order to implement a new fairness-aware recommender model, there are three steps: (1) extend the FairRecommender to implement the model details, and (3) extend the FairTrainer to customize the training process.

Fairness recommendation. For running a meta learning based fairness model, one should (1) indicate the parameters about the environment, data, trainer, evaluation and model by a YAML file, and (2) specify the model and dataset, and (3) launch the program with the quick-start script. An example to train and evaluate FOCF [26] is shown in Figure 2(c). In order to implement a new fairness-aware recommender model, there are three steps: (1) extend the FairTrainer to implement the model details, and (3) extend the FairAbstractMetric to implement the fairness metrics.

Debiased recommendation. To run an implemented debiased recommender model, one needs to follow two steps: (1) specify the setting of the model, dataset, training and evaluation processes via a YAML file, and (2) launch the program by indicating the model and dataset. An example of running PDA [29] based on Yahoo!R3 is shown in Figure 2(d). For implementing a model, one can implement the model architecture by extending the DebiasedRecommender class, and implement the trainer and sampler.

Table 1: The included packages and the implemented models in each package. Due to the page limitation, we omit the references of these models, which can be found in our documents.

| Module        | Package                                      | Models                                          |
|---------------|----------------------------------------------|------------------------------------------------|
| Data          | Data augmentation (RecBole-DA)               | CL4SRec, DuoRec, MMInfoRec, CauseRec, CASR, CCL, CoSeRec |
|               | Meta recommendation (RecBole-MetaRec)        | MeLU, MAMO, LWA, NL, TaNP, MetaEmb, MWU         |
|               | Debias recommendation (RecBole-Debias)       | MF-IPS, PDA, MACR, DICE, CausE, Rel-MF          |
|               | Fairness recommendation (RecBole-FairRec)    | FOCF, PFRC, FairGo, NFCF                        |
|               | Cross-domain recommendation (RecBole-CDR)    | CMF, CFM, DTCR, DeepAPF, NATR, CoNet, BiTWC, EMICR, SSCR, DCDRR |
| Model         | Graph based recommendation (RecBole-GNN)     | NGCF, LightGCN, SGL, HMLET, NCL, SimGCL, SR-GNN, GC-SAN, NISER, LESSR, TAGNN, GCE-GNN, SGN-HN, DiffNet, MHCN, SEFT |
|               | Transformer based recommendation (RecBole-TRM)| TiSASRec, SSE-PT, LightSANs, gMLP, CORE, NRMS, NAML, NPA |
| Others        | Person-job fit (RecBole-PJF)                 | PJFNN, APJFNN, BPJFNN, PJF, PJF, SHP, LPF |

Cross-domain recommendation (RecBole-CDR). Data distribution shift often occurs in cross-domain recommendation. To conduct effective cross-domain models, we consider three representative categories of methods, including collective matrix factorization, representation sharing or combination and knowledge transfer or mapping. Cross-domain recommendation is not a new topic, but few packages have a good coverage of various representative methods in this field.

3.2 Model-oriented Packages

GNN based recommendation (RecBole-GNN). Recently, graph neural networks (GNNs) [9, 22] have been shown effective to model graph structures of various data types, e.g., recommender systems [25]. We implement three types of GNN models tailored to different tasks, including general recommendation, sequential recommendation and social recommendation. Following the original data format in RecBole, we design one new kind of atomic file with suffix .net for structured data (e.g., social network) modeling.

Transformer based recommendation (RecBole-TRM). Another major model architecture we implement is the widely used Transformer and its variants. We implement this package by considering two major tasks, namely sequential recommendation and news recommendation. Although the basic Transformer architectures used in these tasks are similar, the inputs of them are different, which makes Transformer play different roles in sequence modeling. Specifically, in sequential recommendation, Transformer is utilized to capture user behavior correlations; while, in news recommendation, Transformer is utilized to encode text semantics.

Person-job fit recommendation (RecBole-PJF). Besides, we also include a package tailored for the task of person-job fit [6]. This task is an important application and draws much attention from both research and industry communities. This package includes three categories of models, including collaborative filtering models, content-based models and hybrid models. This package designs special data mechanism to support additional data types.

Package summary. Note that each package can be run as an individual project (including the entire pipeline to produce comparison results) but with unified implementation based on RecBole.

4 PACKAGE USAGE

Data augmentation. In our library, the data augmentation package is provided through a series of augmentation interfaces or commands. For invoking the data augmentation package, one just needs to follow two steps, that is, (1) indicating the configuration file, and (2) running the augmentation command based on existing or customized model. Figure 2(a) presents an example on how to run SASRec with a data augmentation strategy called "item_crop".

Meta recommendation. For running a meta learning based recommender, one can use the quick start wrapper (see Figure 2(b)), which will automatically conduct model configuration, trainer configuration, dataset preparation, model training and model evaluation. To implement a new meta learning model, the users can follow three steps: (1) indicate the configuration file, (2) extend the MetaRecommender to implement the model details, and (3) extend the MetaTrainer to customize the training process.

Fairness recommendation. For running an implemented fairness model, one should (1) indicate the parameters about the environment, data, trainer, evaluation and model by a YAML file, and (2) specify the model and dataset, and (3) launch the program with the quick-start script. An example to train and evaluate FOCF [26] is shown in Figure 2(c). In order to implement a new fairness-aware recommender model, there are three steps: (1) extend the Trainer to indicate the specific training process, (2) extend the FairRecommender to implement the model, and (3) extend the FairAbstractMetric to implement the fairness metrics.
Cross-domain recommendation. To run a cross-domain recommender model, the users can set the formatted dataset as either source domain or target domain by configurations and run the model with simple commands. We present an example of how to run EMCDR [17] in our library in Figure 2(e), where we can set the source and target domain datasets, indicate the training modes and so on. For implementing a new algorithm, one can firstly extend the CrossdomainRecommender class and then specify the training modes with simple configurations.

GNN based recommendation. For running the implemented GNN models, there are two major steps to prepare: (1) indicate the customized configuration values and store them in an additional YAML file; (2) specify the model and dataset and launch with the quick-start script. An example to train and evaluate NCL [15] on Movielens-1M dataset with customized configuration is shown in Figure 2(f). For implementing new GNN-based recommendation models, one can reuse or adapt GNN layers for fast reproduction. For example, we can reuse LightGCNConv layer in RecBole-GNN to reproduce GNN-based collaborative filtering models, or replace the graph convolutional layer to new GNN components for further exploration.

Transformer based recommendation. To run an implemented Transformer-based model, one can follow three steps: (1) specify the configuration via a YAML file; (2) indicate the dataset and model; (3) run the model with the quick-start script. An example for running TiSASRec [13] based on Movielens-1M dataset is shown in Figure 2(g). To implement a new Transformer-based recommendation model, the users can reuse Transformer layers implemented by RecBole, or add new implementations in Transformer layers.

Person-job recommendation. This package mostly follows the use of the overall RecBole library, with some adoptions or new interfaces. The steps for developing new models are as follows: create a new model file (e.g., PJFNN.py), implement corresponding functions and save the hyper parameters to the configuration file. Specially, we introduced a new parameter biliteral to control whether the evaluation is for biliteral or not. If it is set to true, the evaluation will be conducted at both candidate and employer sides on the same interaction records.

5 DISCUSSION AND CONCLUSIONS

The included extensions are developed based on a popular recommendation library RecBole, which originally contains more than 70 recommendation models covering the tasks of general recommendation, context-aware recommendation, sequential recommendation and knowledge-based recommendation. Since 2020, RecBole has received extension attention and use on GitHub, about 2K stars and 379 forks till August 15, 2022. In original RecBole, we focus on the design of underlying data structures, general evaluation pipeline and classical recommendation models.

With the rapid progress of recommender systems, we receive increasing requests from RecBole users on the support for up-to-date advances (e.g., debiased, fairness and GNNs). Meanwhile, our team members are also conducting research on these emerging topics or models. Therefore, we develop and release this extended library for enhancing RecBole by incorporating the support on recent advances of recommender systems. Specifically, in this extension, we release eight packages consisting of 65 newly implemented models, and also provide corresponding interfaces for data preparation, model running (with the well-tuned parameters) and evaluation. We believe this extension is a significant contribution to RecBole, which is a valuable resource to the research community. The RecBole team will continually improve this project, making it up-to-date, comprehensive, and flexible for research.

ACKNOWLEDGMENTS

This work is supported in part by National Natural Science Foundation of China No. 61872369 and No. 62102420, Beijing Outstanding Young Scientist Program No. BJWZYJH012019100020098, Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the “Double-First Class” Initiative, Renmin University of China, and Public Computing Cloud, Renmin University of China.
REFERENCES

[1] Vito Walter Ameli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitlesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, and Tommaso Di Noia. 2011. Elliot: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation. In SGIR.

[2] Andreas Argyriou, Miguel González-Fierro, and Le Zhang. 2020. Microsoft Recommenders: Best Practices for Production-Ready Recommendation Systems. In WWW.

[3] Xiangnan He, Yixuan Wang, Bin Wu, Zhonghuan Sun and Jonathan Staniforth. 2017. NeurRec: https://github.com/wubinzzu/NeurRec (2017).

[4] Luwei Chen, Hande Dong, Xiang Wang, Ful Ferr, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. arXiv preprint arXiv:2010.03240 (2020).

[5] Arthur da Costa, Eduardo Fressato, Fernando Neto, Marcelo Mambato, and Ricardo Campello. 2018. Case recommender: a flexible and extensible python framework for recommender systems. In RecSys.

[6] Jeffrey R Edwards. 1991. Person-job fit: A conceptual integration, literature review, and methodological critique. John Wiley & Sons.

[7] Chongming Gao, Shijun Li, Wenqiang Lei, Biao Li, and Peng Jiang. 2022. KaaiRec: A Fully-observed Dataset for Recommender Systems. arXiv preprint arXiv:2202.10842 (2022).

[8] Yingqiang Ge, Shuchang Liu, Ruyouan Gao, Yikun Xian, Yunli Li, Xionganchuao, Petro Liö, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In NIPS.

[9] Nicolas Hug. 2020. Surprise: A Python library for recommender systems. Journal of Open Source Software (2020).

[10] Rashidul Islam, Kamrun Naher Keya, Ziqian Zeng, Shimei Pan, and James Foulds. 2021. Debiasing Career Recommendations with Neural Fair Collaborative Filtering. In TheWebConf.

[11] Koki Iimori, Yukiya Morikawa, Shota Seto, and Jun Sakuma. 2020. A comprehensive evaluation framework for personalized recommendation. In ICDM.

[12] Ruochen Li, Yujie Wang, and Julian J. McAuley. 2020. Towards Personalized Fairness Based on Causal Notion. In RecSys.

[13] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 2019.

[14] Zhan Zhang, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In RecSys.

[15] Shuren Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: A Survey. ACM Comput. Surv. (2022).

[16] Silicon Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In NIPS.

[17] Evaggelia Pitoura, Kostas Stefanidis, and Georgios Koutrika. 2022. Fairness in rankings and recommendations: an overview. VLDB J. (2022).

[18] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In ICML.

[19] Zhu Sun, Di Yu, Hua Feng, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Towards Long-Term Fairness in Recommendation. In TheWebConf. CoRR abs/1903.14000 (2021).

[20] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 2019.

[21] Kuang Zhang, Rui Li, Hong Wang, and Longdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In SIGIR.

[22] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In ICLR.

[23] Ngiao, Kostas Stefanidis, and Georgia Koutrika. 2022. Fairness in recommendation: An embedding and mapping approach.. In IJCAI.

[24] Evaggelia Pitoura, Kostas Stefanidis, and Georgios Koutrika. 2022. Fairness in rankings and recommendations: an overview. VLDB J. (2022).

[25] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In ICML.

[26] Zhu Sun, Di Yu, Hua Feng, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In RecSys.

[27] Shuren Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: A Survey. ACM Comput. Surv. (2022).

[28] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In NIPS.

[29] Meike Zehlike, Kr Yang, and Julia Stoyaovitch. 2021. Fairness in ranking: A survey. arXiv preprint arXiv:2102.14000 (2021).

[30] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: A Survey. ACM Comput. Surv. (2022).

[31] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 2019.

[32] Yang Zhang, Rui Li, Hong Wang, and Longdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In SIGIR.

[33] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In ICLR.

[34] Evaggelia Pitoura, Kostas Stefanidis, and Georgios Koutrika. 2022. Fairness in rankings and recommendations: an overview. VLDB J. (2022).

[35] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In ICML.

[36] Zhu Sun, Di Yu, Hua Feng, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In RecSys.

[37] Shuren Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: A Survey. ACM Comput. Surv. (2022).

[38] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In NIPS.

[39] Meike Zehlike, Kr Yang, and Julia Stoyaovitch. 2021. Fairness in ranking: A survey. arXiv preprint arXiv:2102.14000 (2021).

[40] Shiwei Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: A Survey. ACM Comput. Surv. (2022).

[41] Yang Zhang, Rui Li, Hong Wang, and Longdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In SIGIR.

[42] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In ICLR.

[43] Evaggelia Pitoura, Kostas Stefanidis, and Georgios Koutrika. 2022. Fairness in rankings and recommendations: an overview. VLDB J. (2022).

[44] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In ICML.

[45] Zhu Sun, Di Yu, Hua Feng, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In RecSys.

[46] Shuren Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph Neural Networks in Recommender Systems: A Survey. ACM Comput. Surv. (2022).

[47] Yang Zhang, Rui Li, Hong Wang, and Longdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In SIGIR.