Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 3, No. 2, 2023

Bias-based Denoising Causal Recommendation Algorithm

Xu Yang

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China.

Abstract: Traditional recommendation algorithms, such as collaborative filtering, make recommendations by learning the relevant relationships between users and items. However, considering only the relationships without considering the underlying causal mechanisms would be unfair, uninterpretable, and would lead to bias. In this paper, we propose bias-based denoising causal recommendation algorithm (BDCR) . First, the method dynamically transforms the explicit user-item feedback into implicit feedback with an embedded representation. Then, a truncation function based on causal inference is constructed to remove false positive noise. In addition, traditional recommendations and denoised causal recommendations are aggregated to obtain predictive scores. Finally, experimental results on two real datasets show that the BDCR algorithm outperforms the classical algorithm in terms of recall and NDCG metrics.

Keywords: Causal; Denoising; Bias; Positive; Recommendation.

1. Introduction

Today's era is an information explosion, and recommendation systems are used to solve the user's exploration problem as an information filtering system, i.e., to find the most suitable one among the huge number of available options, thus saving the user's time and effort. Traditional recommender systems include three categories: collaborative filtering, content-based recommendation and hybrid recommendation [1]. They are based on mining or learning relevant patterns in data. In recent years, researchers are not satisfied with simply discovering correlations between learning users, items, and features, but have started to consider causal relationships. Causality refers to the effect of one event or behavior on other events or behaviors.

Recommender systems combined with causal learning are able to deal with different problems including interpretability, fairness, robustness, and unbiasedness [2]. There have been many studies on various causal recommendations, which deal with different problems in recommendations separately. In 2020, sato [3]et al. identifies user/item features as confounders when estimating causal effects of recommendation. In 2021, wei [4] et al. proposed a model-independent counterfactual inference method to construct multi-task learning to learn the effect of item attributes on prediction results and user attributes on prediction results separately, and thus obtain counterfactual ranking scores using only item attributes, and by deducting the counterfactual from the overall score scores, eliminating the problem of popularity bias from a causal perspective. In 2022, Xu [5] et al. use counterfactual explainable recommendation (CountER), which introduces counterfactual reasoning from causal inference to explainable recommendations. Although the above methods are able to improve the performance of recommendation algorithms through causal inference, they fail to remove noise, such as the effect of false-positive interactions, in the original dataset in a timely manner.

2. Related work

Causal reasoning has been developed from correlation studies and contains two main model frameworks, the first is the Potential Outcomes Framework (POF), also known as Rubin Causal Model (RCM), proposed by Rubin et al. in 1974 [6], and the other is the Structural Causal Model (SCM) [7]. Structural causal framework contains two basic concepts: intervention and counterfactual, intervention methods including front door adjustment, back door adjustment, and potential outcome framework to estimate potential outcomes and treatment effects, including methods such as IPW (Inverse Probability Weighting), and double robustness. As Pearl stated, these two frameworks are logically equivalent [8]. Theorems and assumptions in one framework can be equally translated into the language of the other framework. In 1998, Breese et al [9] applied Bayesian networks in causal inference to personalized recommendation and demonstrated the superiority of Bayesian networks under extensive comparative experiments. In 2020, Zhang et al. integrated the learning of propensity and recommendation models into a multi-task learning framework [10] , which is more advantageous than learning alone. In 2022, Wang et al [11] took a pioneering step in considering exposure bias in sequence recommendation by proposing an IPW-based USR approach (unbiased sequential recommender, unbiased sequential recommendation model) to mitigate the confounding factors in sequence behavior. Despite their good performance in causal modeling, they still do not model noise in complex raw data well, while our approach uses a truncation function to remove noise.

Many previous works have experimentally demonstrated the severity of data noise and its negative impact on recommender systems. Cosley et al [12] showed that only 60% of users maintain their ratings when asked to re-rate the same movie. Amatriain et al [13] showed that compared to noiseless data, recommendation performance with noisy data is is significantly affected, with a difference of about 40% in RMSE. Some work, such as negative experience identification notes the identification of negative experiences in implicit signals, and previous work typically collects feedback from different users (e.g., dwell time, gaze patterns, and more detailed item characteristics) to predict user satisfaction [14]. However, these approaches require

additional feedback and extensive manual marking efforts, e.g., users must tell whether they are satisfied with each interaction, and features rely on manual design. Wen et al [15] suggest using "click-to-complete", "click-to-skip " and "non-click" items to train recommenders. The latter two items are considered as negative, but with different weights. By jointly considering different feedbacks, noise is also removed to some extent, but sometimes this feedback is difficult to obtain. The literature [16] explores denoising without adding information during implicit feedback training, pointing to a new research direction for noise removal without using additional feedback. Although these methods are good at removing noise, they either require additional information or do not consider causality, and this paper designs a denoising method that incorporates causal inference.

3. BDCR Algorithm

3.1. BDCR Algorithm Framework

The BDCR algorithm includes embedding layer, BDCR layer, and prediction layer. The algorithm framework is shown in Figure 1. In embedding layer, the original dataset is processed as implicit feedback input, and the user and item features and eigenvalues are input to obtain the embedding; in the BDCR layer, the traditional rating is first entered into the traditional model to calculate the traditional score, and the false positive noise data is removed according to the rating, and the causal debiasing model is input to aggregate the calculated M-values by adding them to the backbone model; in the prediction layer, the inner product calculates the denoised causal score, interest drift degree, and then the fusion of the traditional algorithm with the BDCR algorithm to derive the final prediction.

3.2. Embedding Layer

The publicly available datasets ML-1M and Amazon-book are explicit rating data, and the rating are both on a scale of 1-5. Since most of the existing recommendation systems use implicit feedback recommendations, they need to be converted, and the conversion chosen for the algorithm in this chapter is shown in Equation 1.

$$y_{ui} = \begin{cases} 1, & R = 5 \\ 0, & else \end{cases} \tag{1}$$

where R denotes the original rating, yui denotes the modified rating used in the training process, i.e., a full rating is regarded as a positive example, otherwise it is regarded as a negative example, and any item that the user is not satisfied with is not given a full rating.

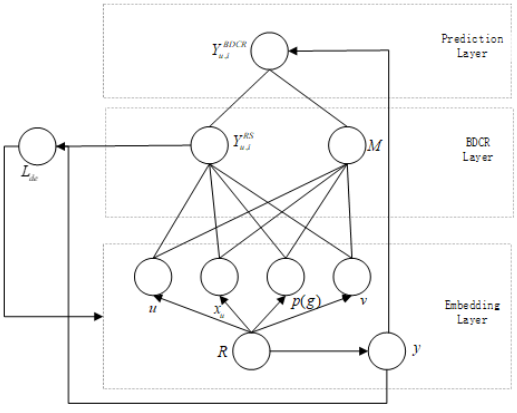

Fig. 1 BDCR algorithm framework

Subsequently, the user features are jointly coded with the item features, with all user feature feature values being 1 and the item feature values being determined according to the category to which the item belongs. Specifically, an item may belong to many categories, and the setting of each item feature value affects the prediction score. However, for information statistics, the more important categories are given priority in the statistics. Take Toy Story in movielens movie dataset as an example, it is firstly an animated movie, then a children's movie, and finally a comedy movie. After comprehensive consideration, the probability feature that Toy Story belongs to the animation category is set to the maximum, while the probability of the children's category and the probability of the comedy category are set to decrease in that order.

3.3. BDCR layer

3.3.1. Removal of false positives

Following the literature [16] , the training loss values for false positives are larger, and the loss values are used to distinguish between true and false positive interactions in the data by setting the loss values as follows.

$$L_{de}(u,i) = \begin{cases} 0, & L(u,i) > \tau \wedge y_{ui} = 1 \\ LCE(u,i), & \text{otherwise} \end{cases} \tag{2}$$

where τ is the threshold value, and since the loss value must decrease with iteration, it is set to a dynamic, varying threshold value with the number of iterations $\tau(t)$. Since the loss value varies across data sets, it is also necessary to set $\tau(t)$ as a function of the discard rate ε :

$$\tau(T) = \min(\alpha T, \epsilon_{\max}) \tag{3}$$

where ε_{max} is the upper discard rate limit, and α is a hyperparameter to adjust the speed to reach the maximum discard rate.

3.3.2. Causal inference with distribution bias

Considering the causal graph shown in Figure 1, there are two paths for user U to influence the matching score: the first path is U→Y, which simply indicates U's true preference for the item; the second path is U→M→Y, which indicates that the user's preference for the category influences the predicted score Y. Since there is a backdoor path U←D→M→Y in the graph, it can be learned that both D and M are confounders, leading to amplification of the spurious bias and requiring intervention to remove the confounding. According to the literature [17], the prediction probability P(Y|U, I) is as follows.

$$\begin{aligned} & P(Y \mid do(U=u), I=i) \\ &= \sum_{d \in \vec{D}} P(d)\, \tilde{f}(u,i,M(d,u)) \\ &\approx f(u,i,M(\sum_{d \in \vec{D}} P(d)d,u)) \end{aligned} \tag{4}$$

where, $f(.)$ is an arbitrary recommended model, M is used as an additional input to the model, and M is calculated as follows.

$$M(\bar{\mathbf{d}}, \mathbf{u}) = \sum_{a=1}^{N} \sum_{b=1}^{K} p(g_a) v_a \odot x_{u,b} \mathbf{u}_b \tag{5}$$

The probability values of the N categories are multiplied by the category representations and then multiplied with the user's feature values and feature representations as dot

products.

3.3.3. Causal Inference Fusion Denoising

Using an implicit feedback denoising method to obtain noisy data in the dataset, interactions with large loss values are dynamically discarded, allowing the dataset to eliminate false positive samples. The data are then fed into the causal debiasing model so that the data are no longer affected by item category bias.

First, the discard rate that increases linearly with the number of iterations is set. After the causal model obtains the prediction scores based on the features and eigenvalues, the prediction scores and the true labels are entered into the following equation.

$$L_{CE} = -\sum_{i=1}^{n} y_i \log \hat{y}_i \tag{6}$$

Next, the LCE is sorted by the principle of smallest to largest to get the corresponding index. The retention rate is obtained from the discard rate, and the index corresponding to the first n loss values is retained. Finally, the predicted values and labels corresponding to the retained indexes are taken out, and the false positive interactions are considered to have been removed, and the cross first loss is recalculated by them, and the recalculated cross first loss is returned to the causal model for back propagation. ALL in all, the false positives are calculated as follows.

$$\mathcal{D} = \underset{\mathcal{D}\subset\mathcal{D}_{pos},|\mathcal{D}|=\lfloor \epsilon(T)*|\mathcal{D}_T| \rfloor}{\arg\max} \sum_{(u,i)\in\mathcal{D}} L_{CE}\left(u,i \mid \Theta_{T-1}\right) \tag{7}$$

where D is the false positive in the positive Dpos and DT is the training data that contains not only Dpos but also the negative sampling data. After the false positives are removed from the training data, denoted as D*, the parameters are updated by the following equation.

$$\theta = \underset{\theta}{\arg\min} \sum_{\left(u,i,y_{u,i}\right)\in\mathcal{D}^*} l\left(f(u,i,M(\bar{d},u)), y_{u,i}\right) \tag{8}$$

3.4. Prediction Layer

Sometimes bias amplification is beneficial to users, so to dynamically adjust the impact of causal inference, scatter is used to measure interest drift in the sequence of user interactions, and thus to adjust the blend of traditional and causal recommendations.

KL scatter is used to measure the difference between two distributions and is equal to a cross-entropy minus an information entropy, but it cannot be used properly in recommender systems due to its asymmetry, i.e., KL(P||Q) ≠ KL(Q||P). JSD (Jensen-Shannon divergence, JS scatter) is also a measure for similarity or difference between two probability distributions as a metric. It can use KL (Kullback-Leibler) scatter to measure its similarity and it can be weakened again in a different dimension.

Since the KL scatter is asymmetric, a slight modification of it can be transformed into a symmetric JS. First, the JS scatter is defined as follows.

$$JSD\left(P \,||\, Q\right) = \frac{1}{2} KL\left(P \,||\, M\right) + \frac{1}{2} KL\left(Q \,||\, M\right) \tag{9}$$

where M=1/2(P+Q) and P and Q denote d_u^1 and d_u^2, respectively, the class distribution of user u on the two segments before and after the interaction sequence. In general, the JS scatter is symmetric and takes values between 0 and 1. The JS scatter and the KL scatter have some similarities but also some significant differences. the JS scatter can consider the overlap between the two probability distributions at the same time and thus calculate their similarity more accurately, whereas the KL scatter treats them as independent probability distributions and thus can only be used to calculate the variability between different probability distributions. After normalizing the JSD:

$$\hat{J}_u = \left(\frac{JSD_u - JSD_{\min}}{JSD_{\max} - JSD_{\min}} \right)^{\alpha} \tag{10}$$

where $\hat{J}_u$ is the normalized value of JSD, $JSD_{\min}$ and $JSD_{\max}$ are the maximum and minimum values of JSD for all users, respectively.

It can be used as a moderating factor to reconcile the traditional model with the BDCR model as shown in Equation 11.

$$Y_{u,i} = \left(1 - \hat{J}_u\right) * Y_{u,i}^{RS} + \hat{J}_u * Y_{u,i}^{BDCR} \tag{11}$$

$Y_{u,i}^{RS}$ is the score predicted by the traditional model, and $Y_{u,i}^{BDCR}$ is the score predicted by the BDCR.

4. Experimental analysis

The experiment focuses on the following questions.

RQ1: How does our method perform compared with baseline methods?

RQ2: How do different hyperparameter affect recommendation performance?

RQ3: How does different sub-model affect model performance?

4.1. Data set and Evaluation

Using the generic datasets ML-1M and Amazon-book, ML-1M is a movie recommendation dataset with rich features, such as user gender and movie genre, grouped by movie genre, and Amazon-book is an Amazon book product dataset grouped by book category, with the dataset description shown in Table 1. For each interaction in the dataset with rating>=5, it is considered as a positive example, and users and items with less than 20 interactions are discarded. The ratio of training set, validation set and test set was 8:1:1 after sorting according to the interaction timestamp of each user. random negative sampling was used as negative samples during the training process.

Evaluation metrics: Since the top-K ranking is used, Recall, NDCG is selected as the evaluation metric. An early stop strategy is adopted, and Recall@10 on the validation set is used as the criterion for selecting the best model.

Table 1. Description of the data set

Dataset	Users	Items	Interactions	Features	Group
ML-1M	3,883	6,040	575,276	13,408	18
Amazon-Book	29115	16,845	1,712,409	46,213	253

4.2. Baseline

The proposed method is generalized and instantiated in two representative recommendation models, FM and NFM, to mitigate noise and bias amplification. The benchmark algorithms for the comparison are as follows.

IPS [18] is the classical approach in causal

recommendation. In order to reduce the weight of items in the multi-array during the debiased training, $P(\boldsymbol{d}u)$ is used here as the user's propensity u, and the propensity clipping technique [33] is used to reduce the propensity variance, where the clipping threshold in {2, 3,... 10} in the search adjustment.

DICE [19] utilizes causal separation of interest and consistent representation embeddings. These embeddings are obtained by training on cause-specific data obtained from causal inference, and each embedding will capture only one cause. Choose $\alpha = 0.1$, $\beta = 0.01$, and the loss function as BPR.

PDA [20] considers that the popularity bias affects items and ratings through exposure and herding respectively, and then uses backdoor adjustment to eliminate the effect on item representation, and finally adds a reasonable popularity to the prediction to achieve full utilization of popularity. The popularity smoothing parameter γ is chosen in steps of 0.02 in {0.02, ..., 0.25}.

DecRS [17] models the causal impact of user representation on prediction scores. An approximation operator for backdoor adjustment is contributed, which can be easily inserted into most recommendation models. Finally, an inference strategy is designed to dynamically adjust the backdoor adjustment according to the user's state.

4.3. Overall performance comparison (RQ1)

Table 2 gives the recommended performance of the compared methods for HR@10, Recall@20, NDCG@10, and NDCG@20. bold indicates the best results, horizontal lines are added under the second-best results, and percentages indicate how much the best results improve performance compared to the second-best results. Overall, BDCR consistently outperforms the other methods across all metrics and all data sets. The performance of FM and NFM was improved under DecRS, DICE and PDA methods, which proves the importance of considering bias on the traditional FM algorithm and NFM algorithm. Moreover, BDCR can be further improved by adding more complex models. However, the IPS has worse performance relative to the ordinary FM and NFM. The possible reason is that the IPS method propensity scores are not accurately obtained and face the effect of high variance, and such propensity scores do not accurately estimate the effect of D on U.

Table 2. Overall performance comparison

Backbone Model	Algorithm	ML-1M				Amazon-book			
		R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
FM	MF/NFM	0.0676	0.1162	0.0566	0.0716	0.0231	0.0409	0.0148	0.0206
	IPS	0.0663	0.1188	0.0556	0.0718	0.0213	0.0369	0.0135	0.0187
	DICE	0.0779	0.1209	0.0575	0.0730	0.0216	0.0377	0.0138	0.0191
	PDA	0.0870	0.1259	0.0589	0.0781	0.0251	0.0455	0.0168	0.0238
	DecRS	0.0888	0.1487	0.0621	0.0799	0.0264	0.0476	0.0170	0.0240
	BDCR	**0.0917**	**0.1489**	**0.0656**	**0.0841**	**0.0273**	**0.0477**	**0.0174**	**0.0242**
	%improve	3.2	0.1	5.6	5.2	3.4	0.2	2.3	0.8
NFM	MF/NFM	0.067	0.1176	0.056	0.0717	0.0228	0.0407	0.0148	0.0207
	IPS	0.0648	0.1135	0.0544	0.0692	0.0213	0.0370	0.0137	0.0189
	DICE	0.0848	0.1143	0.0556	0.0708	0.0206	0.0381	0.0133	0.0190
	PDA	0.0934	0.1333	0.0640	0.0807	0.0215	0.0434	0.0140	0.0197
	DecRS	0.0936	0.1478	0.0654	0.0855	0.0244	0.0401	0.0158	0.0214
	BDCR	**0.094**	**0.1505**	**0.0674**	**0.0857**	**0.0245**	**0.0435**	**0.0161**	**0.0224**
	%improve	0.4	1.8	3.1	0.2	4.9	0.2	1.9	4.6

4.4. Sensitivity of hyper-parameters (RQ2)

The data embedding representation requires a reasonable setting of the item feature values, and in order to verify the impact of different item feature values on the model, the ML-1M dataset is used as an example, with other parameters kept constant, and the probabilities are divided sequentially based on the total number of categories to which the items belong, and a total of five division schemes are verified as shown in Table 3, with a total probability value of 1 for each item, and Table 4 shows the Recall@10, Recall@20, NDCG@10 and NDCG@20 on four indicators.

Table 3. Characteristic values based on the general category of the project

	A class	Two classes	Three categories	Four categories	Five categories	Six categories
Option 1	1	7:3	5:3:2	5:3:1.5:0.5	5:3:1:0.5:0.5	4:3:1: 1:0.5:0.5
Option 2	1	8:2	8:2:0	8:2:0:0	8:2:0: 0:0	8:2:0: 0:0:0
Option 3	1	7: 3	7:2: 1	7:2:1: 0	7:2: 1: 0:0	7:2:1:0: 0:0
Option 4	1	2:8	2:8:0	2:8:0:0	2:8:0: 0:0	2:8:0: 0:0:0
Option 5	1	1:0	1:0:0	1:0:0: 0:0	1:0:0: 0:0:0	1:0:0: 0:0:0

Table 4. Indicators for each program

ML-1M	R@10	R@20	N@10	N@20
Option 1	0.0871	0.1426	0.0629	0.0805
Option 2	0.0911	0.1458	0.0656	0.0828
Option 3	0.0917	0.1489	0.0656	0.0841
Option 4	0.0935	0.1458	0.0659	0.0827
Option 5	0.0948	0.1498	0.0671	0.085

From the table 4, we can see that option 5 works best, i.e., the first category characteristic value of all items is set to 1. However, it will not be set this way in practice, and focusing only on the first category will lose the original meaning of multiple categories. Option 3 and option 4 have their advantages and disadvantages, but most of the indicators in option 3 are better than option 4, and option 3 will focus on more classes, which helps to discover the causality, so option 3 is chosen as the final setting in this work.

4.5. Ablation experiment (RQ3)

Ablation experiments are performed by partially or completely destroying specific components in order to assess their contribution to the overall system. In the case of the present algorithm, it mainly includes a denoising computational approach for false positive feedback and a causal inference for bias amplification, so two different ablation algorithms are designed for these two components, and the backbone model for both applications is chosen to be FM.

FM: As Baseline.

BDCR/C: After data denoising, the causal model is not considered and sent back to FM directly for calculation.

BDCR/D: On the basis of BDCR without considering the effect of denoising on data purity, this algorithm does not truncate according to the loss value after entering the FM model in the calculation, and goes directly to the causal calculation.

Table 5. Comparison of BDCR ablation models

	ML-1M				Amazon-book			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
FM	0.0676	0.1162	0.0566	0.0716	0.0231	0.0409	0.0148	0.0206
BDCR/C	0.0688	0.1205	0.0572	0.073	0.0226	0.0405	0.0148	0.0207
BDCR/D	0.0889	0.1478	0.0653	0.0842	0.0273	0.0469	0.0177	0.0243
BDCR	0.0917	0.1489	0.0656	0.0841	0.0273	0.0477	0.0174	0.0242

From the table 5, it can be found that both ablation algorithms are improved compared to FM, and the biggest improvement is BDCR/D, i.e., the elimination of the denoising module does not have as big an effect on the results as the causal module, which indicates the importance of causal reasoning from the side. Despite the relatively high-performance improvement of the causal module, the experimental results were greatly improved by adding denoising, which also indicates that noise in the data has a significant negative effect on the recommendation.

5. Conclusion

The traditional causal inference algorithm is improved and the BDCR algorithm is proposed. Firstly, an implicit feedback to display feedback conversion is used to take an eigenvalue track for embedding to make it conform to the eigenvalue distribution in realistic situations; secondly, a fusion of denoising and causal inference is used to remove the implied false positive feedback in the dataset and the bias amplification in the recommendation process, respectively; subsequently, the traditional recommendation and the proposed algorithm are dynamically combined by measuring the user interest drift; finally, the BDCR algorithm is tested on two Finally, the superiority of BDCR is verified by comparing it with five classical algorithms on two open datasets.

References

[1] DONG Z, WANG Z, XU J, et al. A Brief History of Recommender Systems[Z]. Ithaca: Cornell University Library, arXiv.org, 2022.

[2] CHEN J, DONG H, WANG X, et al. Bias and Debias in Recommender System: A Survey and Future Directions[J]. 2020.

[3] SATO M, TAKEMORI S, SINGH J, et al. Unbiased learning for the causal effect of recommendation: Proceedings of the 14th ACM Conference on Recommender Systems[C], 2020.

[4] WEI T, FENG F, CHEN J, et al. Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System, Ithaca, 2021. Cornell University Library, arXiv.org, 2021;2020.

[5] TAN J, XU S, GE Y, et al. Counterfactual explainable recommendation: Proceedings of the 30th ACM International Conference on Information & Knowledge Management[C], 2021.

[6] RUBIN D B. Estimating causal effects of treatments in randomized and nonrandomized studies.[J]. Journal of educational Psychology, 1974,66(5): 688.

[7] PEARL J. Causality[M]. Cambridge university press, 2009.

[8] PEARL J. Causal inference in statistics: An overview[J]. 2009.

[9] BALL G, BREESE J. Emotion and personality in a conversational character: Proceedings of the Workshop on Embodied Conversational Characters[C], 1998.

[10] ZHANG W, BAO W, LIU X, et al. Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning: Proceedings of The Web Conference 2020[C], 2020.

[11] WANG Z, SHEN S, WANG Z, et al. Unbiased sequential recommendation with latent confounders: Proceedings of the ACM Web Conference 2022[C], 2022.

[12] COSLEY D, LAM S K, ALBERT I, et al. Is seeing believing? How recommender system interfaces affect users' opinions: Proceedings of the SIGCHI conference on Human factors in computing systems[C], 2003.

[13] AMATRIAIN X, PUJOL J M, OLIVER N. I like it... i like it not: Evaluating user ratings noise in recommender systems: User Modeling, Adaptation, and Personalization: 17th International Conference, UMAP 2009, formerly UM and AH, Trento, Italy, June 22-26, 2009. Proceedings 17[C]: Springer, 2009.

[14] LU H, ZHANG M, MA S. Between clicks and satisfaction: Study on multi-phase user preferences and satisfaction for online news reading: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval[C], 2018.

[15] WEN H, YANG L, ESTRIN D. Leveraging post-click feedback for content recommendations: Proceedings of the 13th ACM Conference on Recommender Systems[C], 2019.

[16] WANG W, FENG F, HE X, et al. Denoising Implicit Feedback for Recommendation: WSDM'21:ACM International Conference on Web Search And Data Mining[C], Virtual, Online, Israel: Association for Computing Machinery, Inc, 2021.

[17] WANG W, FENG F, HE X, et al. Deconfounded Recommendation for Alleviating Bias Amplification, Virtual, Online, Singapore: Association for Computing Machinery, 2021.

[18] SCHNABEL T, SWAMINATHAN A, SINGH A, et al. Recommendations as Treatments: Debiasing Learning and Evaluation: ICML 2016[C], 2016.

[19] ZHENG Y, GAO C, LI X, et al. Disentangling user interest and conformity for recommendation with causal embedding, Ljubljana, Slovenia: Association for Computing Machinery, Inc, 2021.

[20] ZHANG Y, FENG F, HE X, et al. Causal Intervention for Leveraging Popularity Bias in Recommendation, Virtual, Online, Canada: Association for Computing Machinery, Inc, 2021.