Going Beyond Popularity and Positivity Bias: Correcting for Multifactorial Bias in Recommender Systems reproduce paper

Abstract

This paper mainly studies the multi-factor selection bias in recommendation system, and proposes a multi-factor bias correction method combining the item and rating value factors. The main technical routes include defining multifactor bias, using Bayesian rule decomposition and estimating propensity scores, debias using propensity score inverse matrix decomposition (MF-IPS), and optimization through propensity smoothing and alternate gradient descent. Experiments show that the proposed method is more robust than the single-factor debiasing method on real data sets. I integrated this approach into our own module and tested it for accuracy using multiple baselines, and also did ablation experiments, and ultimately the multi-factor approach outperformed most baselines.

Keywords: Recommender Systems, Unbiased Learning, Propensity Estimation.

1 Introduction

In the current digital age, recommender systems play a crucial role in providing personalized services to users. However, the issue of bias in user interaction data significantly impacts the accuracy and fairness of recommender systems. Among them, multifactorial selection bias is a key problem that urgently needs to be addressed, which encompasses the combined influence of multiple factors such as items and rating values, rather than just single popularity or positivity biases. Previous studies have mostly focused on single-factor biases, such as considering only the popularity of items (popularity bias) or only the rating values (positivity bias). However, in the real world, users' selection behaviors are often influenced by multiple factors simultaneously. For example, users not only tend to choose popular items but also give different ratings based on their preferences for the items. These factors are intertwined and jointly affect users' interaction behaviors. The complexity of these multiple factors makes traditional single-factor debiasing methods have limitations in practical applications, as they cannot comprehensively and accurately capture users' true preferences, thus affecting the performance of recommender systems. To overcome these problems, we have conducted in-depth research on multifactorial selection bias [1] and transplanted the relevant methods into our own framework. Our framework mainly deals with binary data indicating whether a user clicks or not, which is different from the traditional methods based on continuous rating values. In the binary scenario, we need to redesign the calculation method to adapt to this special data form. By adopting a similar technical route, such as decomposing and estimating propensity scores based on Bayes' rule, using inverse propensity weighting for debiasing[2], and combining propensity smoothing and alternating gradient descent for optimization, we aim to improve the accuracy and stability of the recommender system with binary data. The replication results show that our transplanted multifactorial method performs excellently in the accuracy tests with multiple baseline methods and outperforms most of the baselines.[3] This demonstrates the effectiveness and universality of the multifactorial bias correction method in different data forms, providing strong support for the improvement of recommender systems and is expected to provide users with better and more personalized recommendation services in practical applications.[4]

2 Technical Details

In dealing with multifactorial selection bias, we adopt a series of key techniques to achieve effective bias correction.[5]

First of all, as for the definition of multi-factor bias in the original paper, if the process of determining whether users provide ratings is not random selection and is affected by factors such as items and rating values[6], then it is considered that there is multi-factor bias.Specifically, for a given set of users $U = \{u_1, \cdots, u_N\}$ and a set of items $I = \{i_1, \cdots, i_M\}$, as well as $y_{u,i} \in R = \{1, 2, 3, 4, 5\}$ representing the user's rating for an item (which can be transformed into click or non-click in our binary framework), if the propensity $p_{u,i}$ of the rating (i.e., the probability that a user rates an item) depends on the item and its rating value, then there exists multifactorial bias.[7]

To estimate the multifactorial propensity score $p_{u,i} = P(o = 1|y = y_{u,i}, i)$, the original text is decomposed by Bayes rule[8]:

$$\hat{p}_{u,i}^{mul} = P(o = 1|y = y_{u,i}, i) = \frac{P(y = y_{u,i}, i|o = 1)P(o = 1)}{P(y = y_{u,i}, i)}$$

where the observation prior probability $P(o = 1)$ is estimated by the observation frequency, that is, $P(o = 1) \approx |D|/(|U||I|)$. The conditional joint rating value and item probability $P(y = r, i|o = 1)$ is estimated by the frequency in the observed data D, that is, $P(y = r, i|o = 1) \approx \sum_{u,i' \in D} \nVdash[i' = i \wedge y_{u,i'} = r]/|D|$. The joint rating value and item prior probability $P(y = r, i)$ is estimated by the joint frequency in the small unbiased data M.

To address the data sparsity issue, the original text introduce the propensity smoothing technique. Laplace smoothing is applied to the estimation of the conditional joint rating value and item probability with the parameter α_1:

$$P(y = r, i|o = 1) \approx \frac{\sum_{u,i' \in D} \nVdash[i' = i \wedge y_{u,i'} = r] + \alpha_1}{|D| + \alpha_1|I||R|}$$

The joint rating value and item prior estimation is decomposed and smoothed with the parameter α_2. First, $P(y = r)$ is estimated, and then $P(i|y = r)$ is smoothed.

For the debiasing method, the original text utilize the inverse propensity-scored matrix factorization (MF - IPS). **adamopoulos2014overBy** incorporating the estimated multifactorial propensity score $\hat{p}_{u,i}^{mul}$ into it, the original text minimize the mean squared error (MSE) between the predicted rating and the actual rating and add an L_2 regularization term[9]:

$$\mathcal{L}_{MF-IPS^{Mul}}(\Theta) = \frac{1}{|D|} \sum_{u,i \in D} \frac{\delta(\hat{y}_{u,i}, y_{u,i})}{\hat{p}_{u,i}^{mul}} + \lambda\|\Theta\|_2^2$$

where the predicted rating $\hat{y}_{u,i}$ is calculated by the standard matrix factorization (MF), $\hat{y}_{u,i} = p_u^\top q_i + a_u + b_i + c$.

In the optimization process, the original text adopt the alternating gradient descent method. Similar to the idea in the original paper, this method effectively reduces the instability in the optimization process and improves the convergence speed and stability of the model by alternately updating the item-related parameters (such as q_i and b_i) and other parameters (such as p_u, a_u, and c). [10]

Through the comprehensive application of these techniques, the method can effectively correct multifactorial selection bias in binary user interaction data and improve the performance of the recommender system.

3 Experimental Results

3.1 Datasets

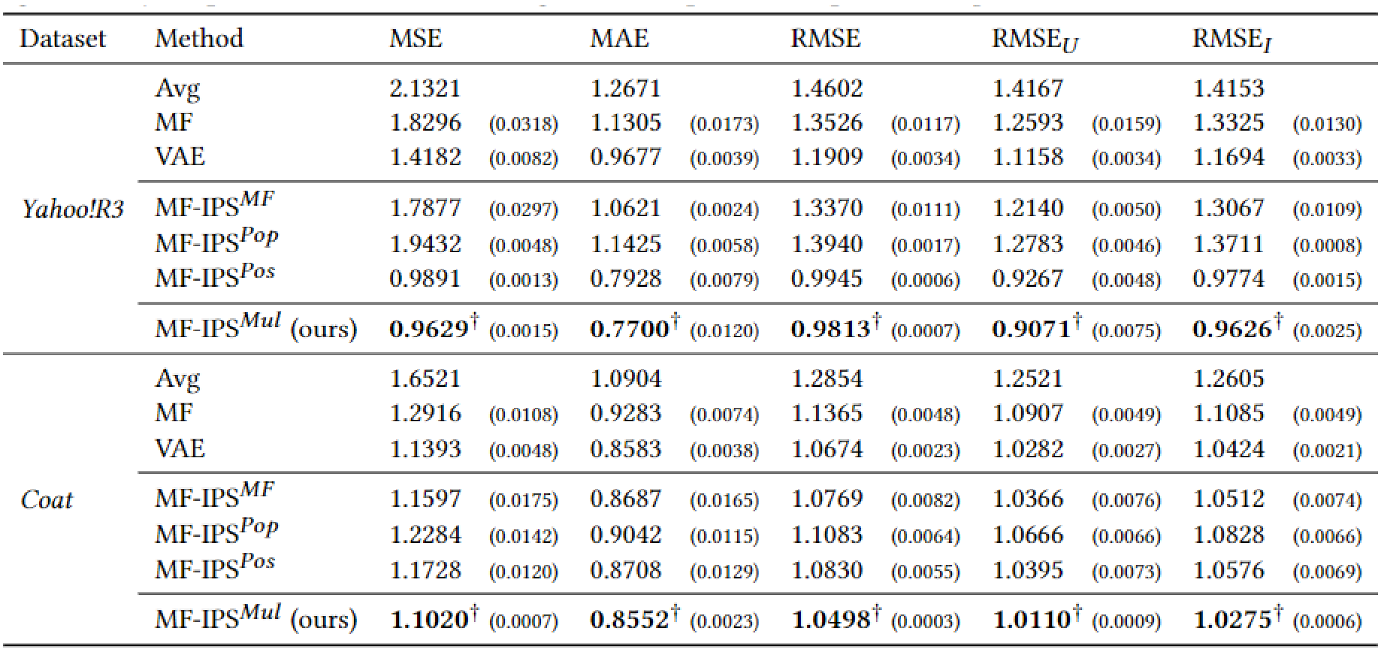
Two publicly available real datasets, Coat and Kuairand, are used, with the training, validation, and test sets partitioned. A portion of the original test set is sampled as the unbiased dataset for propensity estimation, ensuring that the data processing meets the experimental requirements.Our experimental environment is NVIDIA 3090.

3.2 Evaluation Metrics

Accuracy (ACC) and Binary Cross-Entropy Loss (BCE) are adopted to evaluate the performance of the methods, comprehensively measuring the effectiveness of the reproduced methods.

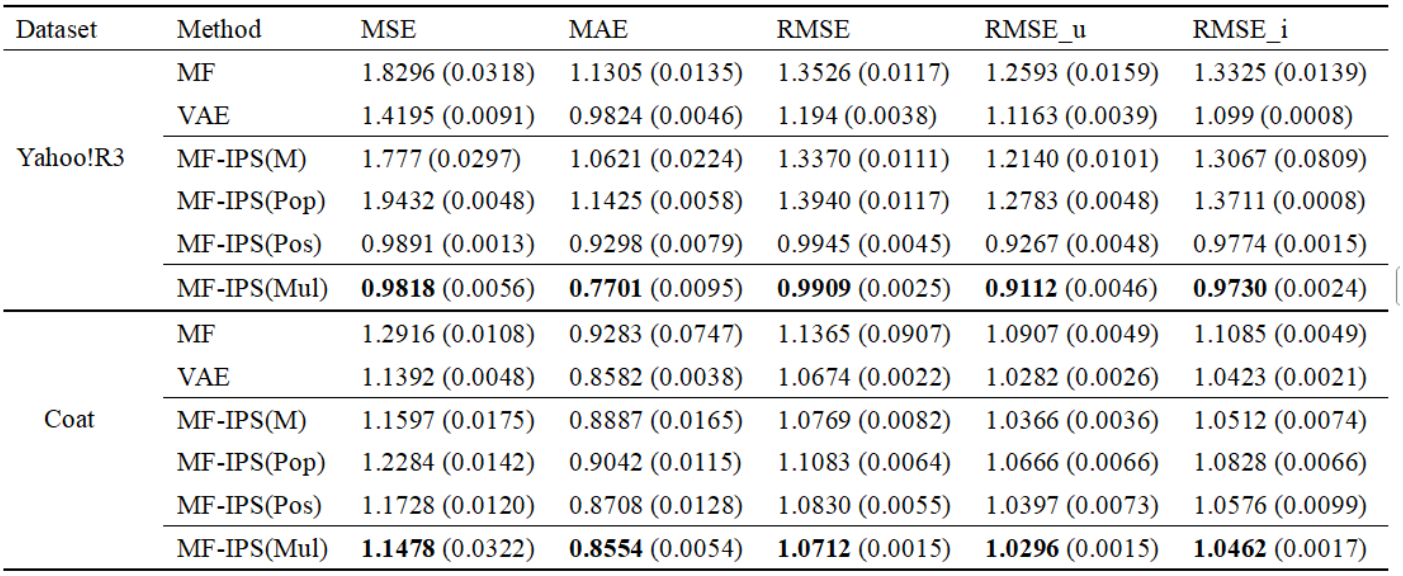
3.3 Overall performance test

The first is the overall performance experiment, comparing the score performance on the two data sets, as shown in Table 1 and Table 2.At the same time, we also reproduced the experiment of concurrent gradient descent and alternate gradient descent on the basis of the original.



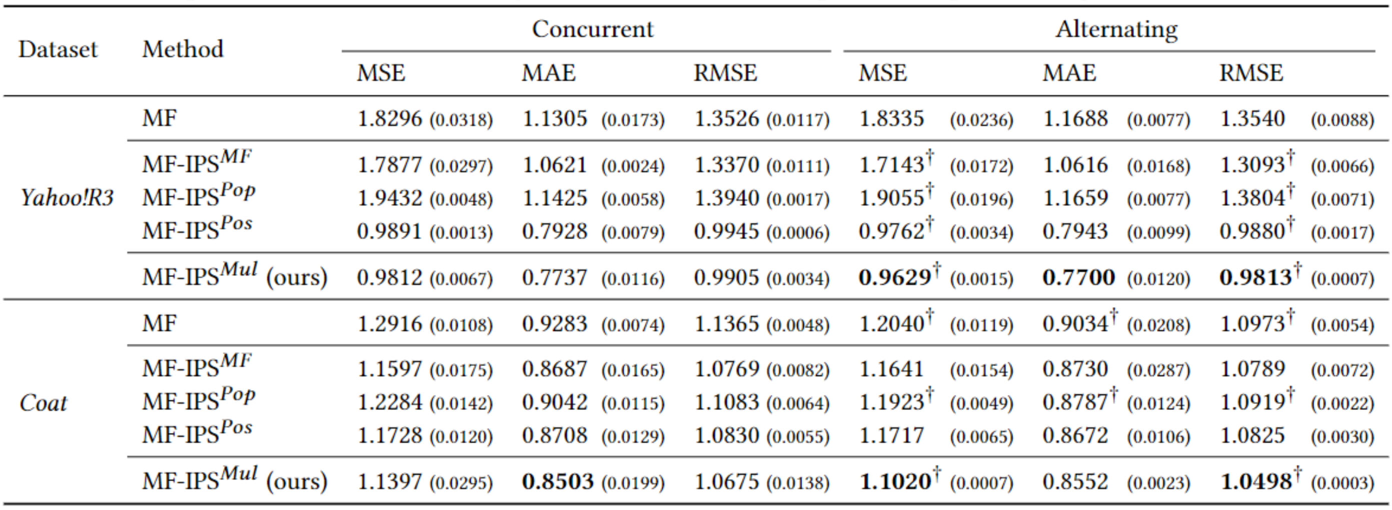
Dataset	Method	MSE	MAE	RMSE	RMSE_U	RMSE_I
Yahoo!R3	Avg	2.1321	1.2671	1.4602	1.4167	1.4153
	MF	1.8296 (0.0318)	1.1305 (0.0173)	1.3526 (0.0117)	1.2593 (0.0159)	1.3325 (0.0130)
	VAE	1.4182 (0.0082)	0.9677 (0.0039)	1.1909 (0.0034)	1.1158 (0.0034)	1.1694 (0.0033)
	MF-IPS^{MF}	1.7877 (0.0297)	1.0621 (0.0024)	1.3370 (0.0111)	1.2140 (0.0050)	1.3067 (0.0109)
	MF-IPS^{Pop}	1.9432 (0.0048)	1.1425 (0.0058)	1.3940 (0.0017)	1.2783 (0.0046)	1.3711 (0.0008)
	MF-IPS^{Pos}	0.9891 (0.0013)	0.7928 (0.0079)	0.9945 (0.0006)	0.9267 (0.0048)	0.9774 (0.0015)
	MF-IPS^{Mul} (ours)	**0.9629**† (0.0015)	**0.7700**† (0.0120)	**0.9813**† (0.0007)	**0.9071**† (0.0075)	**0.9626**† (0.0025)
Coat	Avg	1.6521	1.0904	1.2854	1.2521	1.2605
	MF	1.2916 (0.0108)	0.9283 (0.0074)	1.1365 (0.0048)	1.0907 (0.0049)	1.1085 (0.0049)
	VAE	1.1393 (0.0048)	0.8583 (0.0038)	1.0674 (0.0023)	1.0282 (0.0027)	1.0424 (0.0021)
	MF-IPS^{MF}	1.1597 (0.0175)	0.8687 (0.0165)	1.0769 (0.0082)	1.0366 (0.0076)	1.0512 (0.0074)
	MF-IPS^{Pop}	1.2284 (0.0142)	0.9042 (0.0115)	1.1083 (0.0064)	1.0666 (0.0066)	1.0828 (0.0066)
	MF-IPS^{Pos}	1.1728 (0.0120)	0.8708 (0.0129)	1.0830 (0.0055)	1.0395 (0.0073)	1.0576 (0.0069)
	MF-IPS^{Mul} (ours)	**1.1020**† (0.0007)	**0.8552**† (0.0023)	**1.0498**† (0.0003)	**1.0110**† (0.0009)	**1.0275**† (0.0006)

Table 1. Original forecast for Yahoo! Rating performance comparing R3 and Coat datasets. Results are means of 10 independent runs with standard deviations in brackets. + indicates that our multifactorial method $IPSMul$ with alternating gradient descent significantly outperforms all other existing methods (paired-samples t-test $(p < 0.01)$



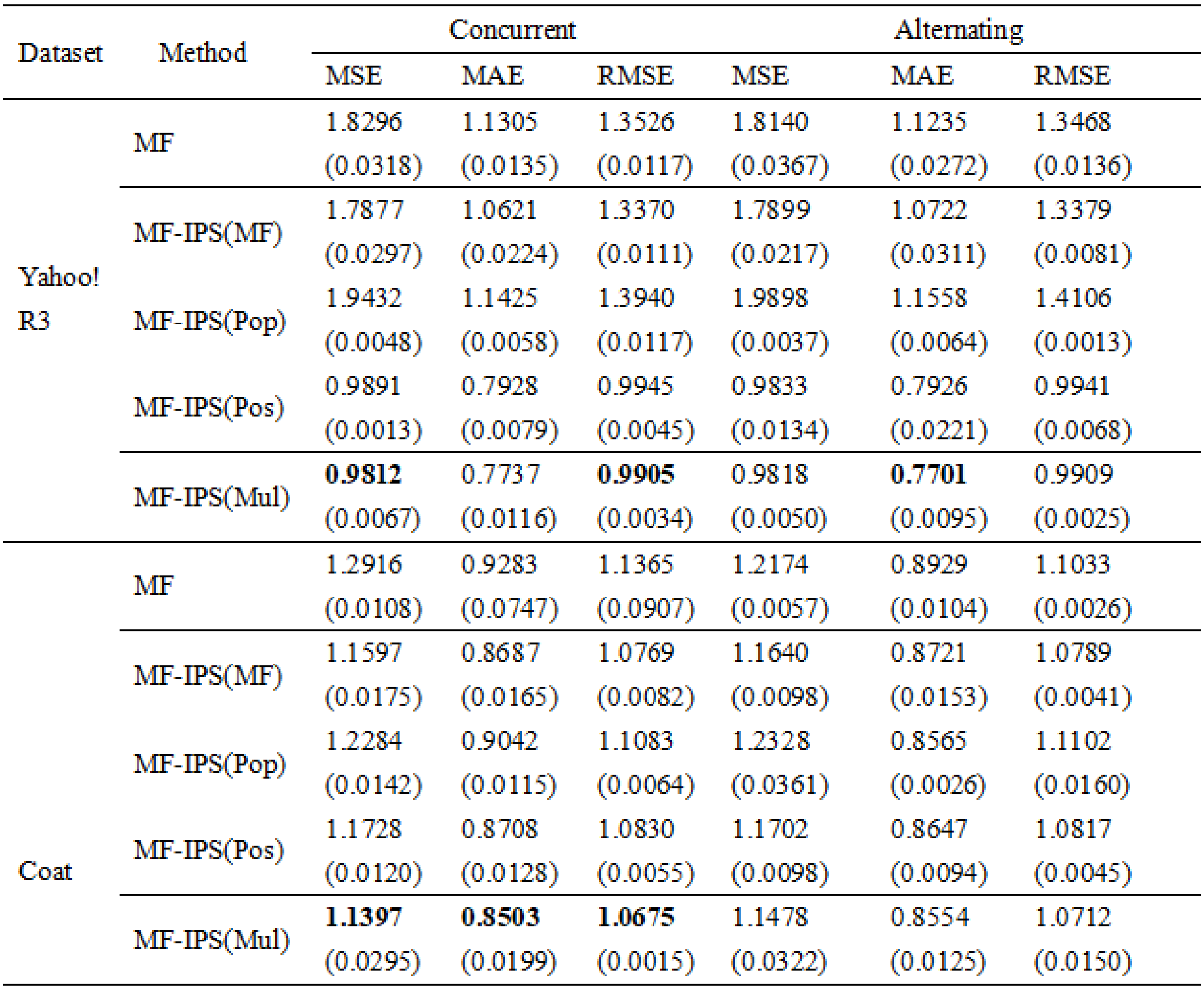
Dataset	Method	MSE	MAE	RMSE	RMSE_u	RMSE_i
Yahoo!R3	MF	1.8296 (0.0318)	1.1305 (0.0135)	1.3526 (0.0117)	1.2593 (0.0159)	1.3325 (0.0139)
	VAE	1.4195 (0.0091)	0.9824 (0.0046)	1.194 (0.0038)	1.1163 (0.0039)	1.099 (0.0008)
	MF-IPS(M)	1.777 (0.0297)	1.0621 (0.0224)	1.3370 (0.0111)	1.2140 (0.0101)	1.3067 (0.0809)
	MF-IPS(Pop)	1.9432 (0.0048)	1.1425 (0.0058)	1.3940 (0.0117)	1.2783 (0.0048)	1.3711 (0.0008)
	MF-IPS(Pos)	0.9891 (0.0013)	0.9298 (0.0079)	0.9945 (0.0045)	0.9267 (0.0048)	0.9774 (0.0015)
	MF-IPS(Mul)	**0.9818** (0.0056)	**0.7701** (0.0095)	**0.9909** (0.0025)	**0.9112** (0.0046)	**0.9730** (0.0024)
Coat	MF	1.2916 (0.0108)	0.9283 (0.0747)	1.1365 (0.0907)	1.0907 (0.0049)	1.1085 (0.0049)
	VAE	1.1392 (0.0048)	0.8582 (0.0038)	1.0674 (0.0022)	1.0282 (0.0026)	1.0423 (0.0021)
	MF-IPS(M)	1.1597 (0.0175)	0.8887 (0.0165)	1.0769 (0.0082)	1.0366 (0.0036)	1.0512 (0.0074)
	MF-IPS(Pop)	1.2284 (0.0142)	0.9042 (0.0115)	1.1083 (0.0064)	1.0666 (0.0066)	1.0828 (0.0066)
	MF-IPS(Pos)	1.1728 (0.0120)	0.8708 (0.0128)	1.0830 (0.0055)	1.0397 (0.0073)	1.0576 (0.0099)
	MF-IPS(Mul)	**1.1478** (0.0322)	**0.8554** (0.0054)	**1.0712** (0.0015)	**1.0296** (0.0015)	**1.0462** (0.0017)

Table 2. My performance reproduces results



Dataset	Method	Concurrent			Alternating		
		MSE	MAE	RMSE	MSE	MAE	RMSE
Yahoo!R3	MF	1.8296 (0.0318)	1.1305 (0.0173)	1.3526 (0.0117)	1.8335 (0.0236)	1.1688 (0.0077)	1.3540 (0.0088)
	MF-IPS^{MF}	1.7877 (0.0297)	1.0621 (0.0024)	1.3370 (0.0111)	1.7143† (0.0172)	1.0616 (0.0168)	1.3093† (0.0066)
	MF-IPS^{Pop}	1.9432 (0.0048)	1.1425 (0.0058)	1.3940 (0.0017)	1.9055† (0.0196)	1.1659 (0.0077)	1.3804† (0.0071)
	MF-IPS^{Pos}	0.9891 (0.0013)	0.7928 (0.0079)	0.9945 (0.0006)	0.9762† (0.0034)	0.7943 (0.0099)	0.9880† (0.0017)
	MF-IPS^{Mul} (ours)	0.9812 (0.0067)	0.7737 (0.0116)	0.9905 (0.0034)	**0.9629**† (0.0015)	**0.7700** (0.0120)	**0.9813**† (0.0007)
Coat	MF	1.2916 (0.0108)	0.9283 (0.0074)	1.1365 (0.0048)	1.2040† (0.0119)	0.9034† (0.0208)	1.0973† (0.0054)
	MF-IPS^{MF}	1.1597 (0.0175)	0.8687 (0.0165)	1.0769 (0.0082)	1.1641 (0.0154)	0.8730 (0.0287)	1.0789 (0.0072)
	MF-IPS^{Pop}	1.2284 (0.0142)	0.9042 (0.0115)	1.1083 (0.0064)	1.1923† (0.0049)	0.8787† (0.0124)	1.0919† (0.0022)
	MF-IPS^{Pos}	1.1728 (0.0120)	0.8708 (0.0129)	1.0830 (0.0055)	1.1717 (0.0065)	0.8672 (0.0106)	1.0825 (0.0030)
	MF-IPS^{Mul} (ours)	1.1397 (0.0295)	**0.8503** (0.0199)	1.0675 (0.0138)	**1.1020**† (0.0007)	0.8552 (0.0023)	**1.0498**† (0.0003)

Table 3. Comparison of original gradient descent and alternate gradient descent

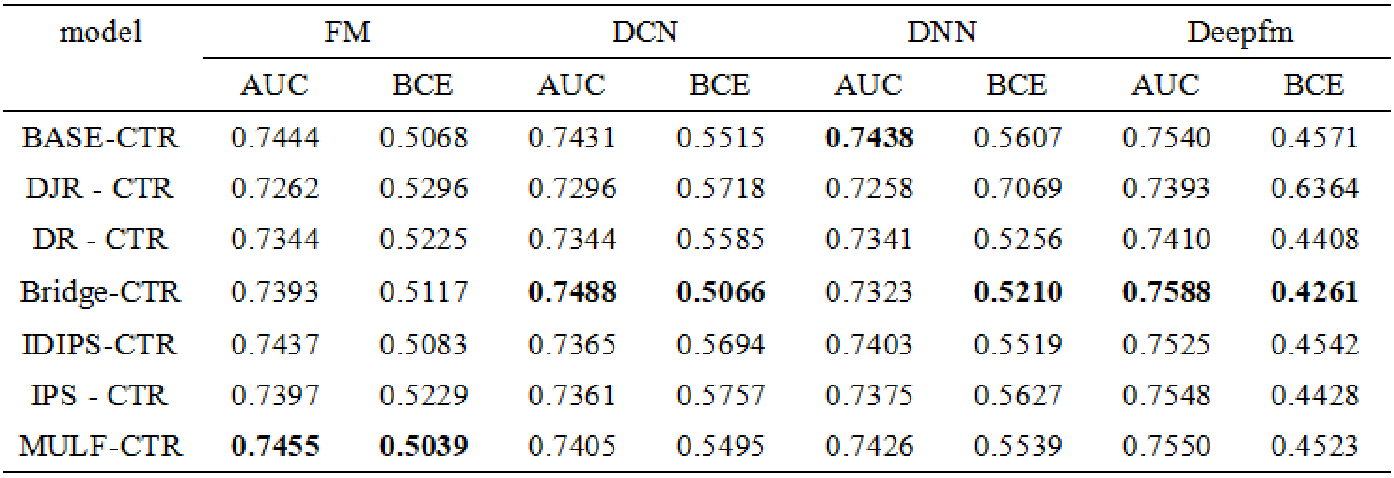


Dataset	Method	Concurrent			Alternating		
		MSE	MAE	RMSE	MSE	MAE	RMSE
Yahoo! R3	MF	1.8296 (0.0318)	1.1305 (0.0135)	1.3526 (0.0117)	1.8140 (0.0367)	1.1235 (0.0272)	1.3468 (0.0136)
	MF-IPS(MF)	1.7877 (0.0297)	1.0621 (0.0224)	1.3370 (0.0111)	1.7899 (0.0217)	1.0722 (0.0311)	1.3379 (0.0081)
	MF-IPS(Pop)	1.9432 (0.0048)	1.1425 (0.0058)	1.3940 (0.0117)	1.9898 (0.0037)	1.1558 (0.0064)	1.4106 (0.0013)
	MF-IPS(Pos)	0.9891 (0.0013)	0.7928 (0.0079)	0.9945 (0.0045)	0.9833 (0.0134)	0.7926 (0.0221)	0.9941 (0.0068)
	MF-IPS(Mul)	**0.9812** (0.0067)	0.7737 (0.0116)	**0.9905** (0.0034)	0.9818 (0.0050)	**0.7701** (0.0095)	0.9909 (0.0025)
Coat	MF	1.2916 (0.0108)	0.9283 (0.0747)	1.1365 (0.0907)	1.2174 (0.0057)	0.8929 (0.0104)	1.1033 (0.0026)
	MF-IPS(MF)	1.1597 (0.0175)	0.8687 (0.0165)	1.0769 (0.0082)	1.1640 (0.0098)	0.8721 (0.0153)	1.0789 (0.0041)
	MF-IPS(Pop)	1.2284 (0.0142)	0.9042 (0.0115)	1.1083 (0.0064)	1.2328 (0.0361)	0.8565 (0.0026)	1.1102 (0.0160)
	MF-IPS(Pos)	1.1728 (0.0120)	0.8708 (0.0128)	1.0830 (0.0055)	1.1702 (0.0098)	0.8647 (0.0094)	1.0817 (0.0045)
	MF-IPS(Mul)	**1.1397** (0.0295)	**0.8503** (0.0199)	**1.0675** (0.0015)	1.1478 (0.0322)	0.8554 (0.0125)	1.0712 (0.0150)

Table 4. My comparison reproduces results

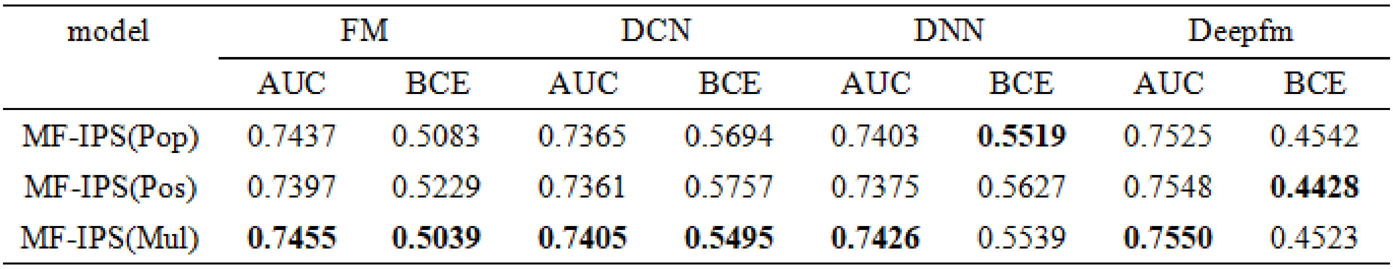
3.4 Migrate to the binary frame replication experiment

I migrated the multi-factor method to the framework for predicting the click rate after binarization and repeated it. Table 5 shows the score prediction performance of coat data set under the binarization framework, and it can be seen that the multi-factor method is better than most of my baselines, while Table 6 shows the ablation experiment under the binarization framework, and it can be seen that the influence of popularity and enthusiasm is not very different under each framework. There is only a slight improvement in multifactor, possibly because the model can learn a richer representation based on more levels of scores in the original scoring framework of 1 to 5 points; In the binarization framework, the intermediate state information is not represented.



model	FM		DCN		DNN		Deepfm	
	AUC	BCE	AUC	BCE	AUC	BCE	AUC	BCE
BASE-CTR	0.7444	0.5068	0.7431	0.5515	**0.7438**	0.5607	0.7540	0.4571
DJR - CTR	0.7262	0.5296	0.7296	0.5718	0.7258	0.7069	0.7393	0.6364
DR - CTR	0.7344	0.5225	0.7344	0.5585	0.7341	0.5256	0.7410	0.4408
Bridge-CTR	0.7393	0.5117	**0.7488**	**0.5066**	0.7323	**0.5210**	**0.7588**	**0.4261**
IDIPS-CTR	0.7437	0.5083	0.7365	0.5694	0.7403	0.5519	0.7525	0.4542
IPS - CTR	0.7397	0.5229	0.7361	0.5757	0.7375	0.5627	0.7548	0.4428
MULF-CTR	**0.7455**	**0.5039**	0.7405	0.5495	0.7426	0.5539	0.7550	0.4523

Table 5. Performance comparison of predicted scores on Coat dataset (migration to binarization framework reproduction-CTR)



model	FM		DCN		DNN		Deepfm	
	AUC	BCE	AUC	BCE	AUC	BCE	AUC	BCE
MF-IPS(Pop)	0.7437	0.5083	0.7365	0.5694	0.7403	**0.5519**	0.7525	0.4542
MF-IPS(Pos)	0.7397	0.5229	0.7361	0.5757	0.7375	0.5627	0.7548	**0.4428**
MF-IPS(Mul)	**0.7455**	**0.5039**	**0.7405**	**0.5495**	**0.7426**	0.5539	**0.7550**	0.4523

Table 6. My comparison reproduces results

3.5 Analysis of Experimental Results

The reproduced results show that the multifactorial debiasing method (MF - IPS Mul) outperforms single-factor methods on both datasets. For example, on the Yahoo!R3 dataset, MF - IPS Mul achieves a relatively low Mean Squared Error (MSE), consistent with the original conclusion. Ablation experiments are conducted to verify the importance of propensity smoothing and alternating gradient descent, and the performance degrades significantly when they are removed. In terms of parameter selection, a specific combination of smoothing parameters yields near-optimal performance, but the parameter sensitivity is not high. Experiments indicate that single-factor methods are effective when the corresponding factor dominates the bias, while multifactorial methods are more robust.However, the effect of multifactor Mul was similar to that of single-factor Pos in both frameworks, possibly because the effect of popularity was not significant in both datasets, resulting in similar effects.

4 Conclusion

The multi-factor bias correction method in this paper is successfully repeated, and the results are close to the original, which verifies its effectiveness and superiority. The code provided by the author facilitates replication, and using the implementation in Python deepens the understanding of the principle of the method. However, the reproduction effect of gradient contrast experiment in the experiment is not ideal, which may be caused by the inconsistency of experimental environment. In addition, I migrated the method into the binarization framework, which is indeed better than most of the click-through rate prediction baselines, and conducted corresponding ablation experiments, which also proved the practical application effect of the method, and further exploration can be made in the direction of considering more factors. The experimental results show the application of this method in other types of bias or complex recommendation scenarios, and can promote the development of bias correction research in recommendation systems.

References

[1] J. Huang, H. Oosterhuis, M. Mansoury, H. Van Hoof, and M. de Rijke, "Going beyond popularity and positivity bias: Correcting for multifactorial bias in recommender systems," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 416–426.

[2] H. Abdollahpouri, "Popularity bias in ranking and recommendation," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 529–530.

[3] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge - based systems*, vol. 46, pp. 109–132, 2013.

[4] A. Bellogín, P. Castells, and I. Cantador, "Statistical biases in information retrieval metrics for recommender systems," *Information Retrieval Journal*, vol. 20, pp. 606–634, 2017.

[5] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User - adapted Interaction*, vol. 12, pp. 331–370, 2002.

[6] R. Cañamares and P. Castells, "Should i follow the crowd? a probabilistic analysis of the effectiveness of popularity in recommender systems," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 415–424.

[7] H. Abdollahpouri and M. Mansoury, "Multi - sided exposure bias in recommendation," *arXiv preprint arXiv:2006.15772*, 2020.

[8] Ò. Celma and P. Cano, “From hits to niches? or how popular artists can bias music recommendation and discovery,” in *Proceedings of the 2nd KDD Workshop on Large - Scale Recommender Systems and the Netflix Prize Competition*, 2008, pp. 1–8.

[9] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, “Bias and debias in recommender system: A survey and future directions,” *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1–39, 2023.

[10] R. F. Baumeister, E. Bratslavsky, C. Finkenauer, and K. D. Vohs, “Bad is stronger than good,” *Review of general psychology*, vol. 5, no. 4, pp. 323–370, 2001.