Addressing bias in Recommender Systems: A Case Study on Data Debiasing Techniques in Mobile Games

Yixiong Wang[1,†], Maria Paskevich[1,*,†] and Hui Wang[1]

[1] *King, Malmskillnadsgatan 19, 111 57 Stockholm, Sweden*

Abstract

The mobile gaming industry, particularly the free-to-play sector, has been around for more than a decade, yet it still experiences rapid growth. The concept of games-as-service requires game developers to pay much more attention to recommendations of content in their games. With recommender systems (RS), the inevitable problem of bias in the data comes hand in hand. A lot of research has been done on the case of bias in RS for online retail or services, but much less is available for the specific case of the game industry. Also, in previous works, various debiasing techniques were tested on explicit feedback datasets, while it is much more common in mobile gaming data to only have implicit feedback. This case study aims to identify and categorize potential bias within datasets specific to model-based recommendations in mobile games, review debiasing techniques in the existing literature, and assess their effectiveness on real-world data gathered through implicit feedback. The effectiveness of these methods is then evaluated based on their debiasing quality, data requirements, and computational demands.

Keywords

Recommender systems, In-game recommendation, Debiasing, Mobile games

1. Introduction

In the context of mobile gaming, delivery of content to players through recommendations plays an important role. It could include elements such as, for example, in-game store products or certain parts of content. However, RSs used within this context are susceptible to bias due to (1) limited exposure: unlike in webshops (e.g. Amazon), available placements for sellable products in mobile games are often limited, and showing one product to a user means that alternatives would not be displayed; (2) the common approach of segmenting content through fixed heuristics before adopting RS introduces biases in the training data, which influences the development of these models. Traditionally, at King we have been addressing these biases by either training models on biased data, or by establishing holdout groups of users who would receive random recommendations for a period of time in order to collect a uniform dataset that reflects user preference in an unbiased way. Although the second approach allows the collection of unbiased data, it could compromise user experience for a segment of players, and may lead to significant operational costs and potential revenue losses. In previous studies, researchers have primarily focused on data derived from explicit feedback, where users rate items using a numerical scale, and various debiasing techniques are tested on this data. However, within the realm of mobile gaming, obtaining explicit feedback affects from user experience, making it challenging to collect. As an alternative, data is often collected through implicit feedback [1], where user preferences are inferred from behaviors such as impressions, purchases, and other interactions. Given these challenges, our objectives in this study are: (1) to identify and categorize potential bias within our datasets; (2) to conduct a review of existing literature on debiasing techniques and assess their effectiveness on publicly available datasets; (3) to adapt and apply debiasing strategies, originally developed for explicit feedback data, to the implicit feedback data specific to King, and (4) to evaluate and compare the efficacy of different methods based on the quality of debiasing, data requirements, and computational complexity.

2. Related work

The existing literature on addressing debiasing techniques in RS presents a well-structured and categorized list of methodologies [2][3]. It suggests that the selection of particular debiasing techniques should depend on the specific types of bias present in the data, as well as on the availability of unbiased data samples. In recommender systems for mobile games, various types of bias can arise, including but not limited to selection bias, exposure bias, position bias, and conformity bias. Some of the relevant methods to debias the data in these cases could be The Inverse Propensity Scoring (IPS) [4] method, which deals with selection and exposure biases by weighting observations inversely to their selection probability, and does so without need for unbiased data. Yet the method could potentially result in high variance due to the challenges in accurately estimating propensities. Potential solutions to the high variance issue of IPS method include, for example, using Doubly Robust (DR) learning [5] that introduces a novel approach to loss functions as a combination of IPS-based models with imputation-based models. The combination of two models assures doubly robustness property when either of the two components (propensity estimation or imputed data) remains accurate. This method, though, relies on having an unbiased data sample to work. Another option is model-agnostic and bias-agnostic solutions like AutoDebias [6], which are based on meta-learning to dynamically assign weights within the RS, aiming to neutralize biases across the board. A potential benefit of such solution is that it doesn't require knowing the types of bias present in the data, but as a downside, it also relies on randomized samples. In addition, the process of fitting multiple models makes training more computationally demanding. Despite the advances and variety of available debiasing techniques, applying Recommendation Systems to mobile gaming content remains a relatively untapped

RobustRecSys: Design, Evaluation, and Deployment of Robust Recommender Systems Workshop @ RecSys 2024, 18 October, 2024, Bari, Italy.

[*] Corresponding author.

[†] These authors contributed equally.

wyx.ei.99@gmail.com (Y. Wang); maria.paskevich@king.com (M. Paskevich); maddy.hui.wang@king.com (H. Wang)

0000-0001-8904-2052 (Y. Wang); 0009-0006-6211-1824 (M. Paskevich); 0009-0004-4190-9410 (H. Wang)

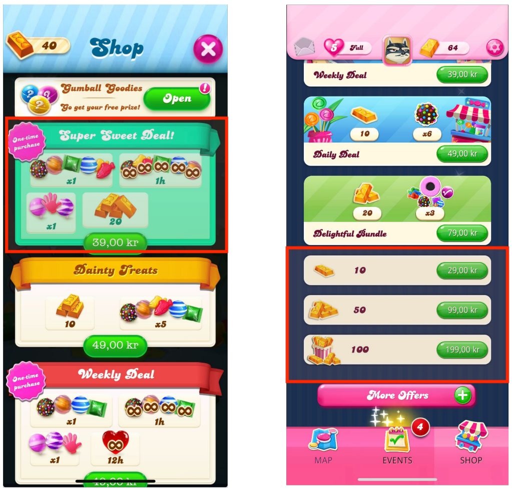

Figure 1: Examples of content placements in Candy Crush Soda Saga (left) and Candy Crush Saga (right), highlighting biases: selection bias with a prominently placed product (left) and exposure bias with limited visibility, where products are hidden behind the "More Offers" button (right).

Table 1
The sizes and feedback types of all datasets used in this study. A key difference is that the open datasets (COAT and YahooR3!) provide explicit feedback, while the proprietary datasets (A, B, and C) offer only implicit feedback (purchase/no purchase). Set A, a proprietary dataset, lacks randomized data, limiting debiasing options.

Dataset	Biased samples	Unbiased samples	Feedback type
COAT	311k	54k	Explicit
yahooR3!	12.5k	75k	Explicit
Set A	47.6k	-	Implicit
Set B	100k	218k	Implicit
Set C	980k	1.2mln	Implicit

area, with most of the publications focusing on building recommendations [7] [8] [9], and not on issues of imbalance and bias. Previous efforts at King introduced DFSNet [10], an end-to-end model-specific debiasing technique that enables training an in-game recommender on an imbalanced dataset without randomized data. This work aims to enrich King's debiasing toolkit by exploring model-agnostic solutions, specifically focusing on the challenges of content recommendations within mobile games. However, the architecture of DFSNet is complex, involving multiple modules, which can make the implementation and maintenance challenging. Moreover, it requires constant feedback loops over time and the model's performance is highly dependent on the quality and recency of the training data.

3. Methodology

3.1. Datasets

Our study utilized two public datasets (COAT[4], yahooR3![13]) to validate theoretical results and three proprietary datasets from King (Set A, Set B, Set C) that are focused on user-item interactions in game shops within Match-3 Game A and Match-3 Game B (Fig.1). The sizes of each dataset, along with their respective feedback types, are provided in Table 1. We aimed to observe the effectiveness of different techniques on datasets collected with explicit feedback (public datasets), and those with implicit feedback (King's datasets). Explicit feedback is typically collected by asking users to rate items on a numerical scale, for example from 1 to 5, where 1 indicates disinterest, 2 signifies dissatisfaction, and 5 shows a preference. In contrast, Implicit feedback (as in the proprietary datasets) involves a binary response from users: purchase or non-purchase. This setup makes it harder to accurately measure user preferences. As discussed in the Introduction, mobile games often have limited space for displaying sellable products, which is the case for all three proprietary datasets. This limitation leads to exposure bias in the data. Additionally, placement of different products within the game shop creates positional bias, with some items displayed in more appealing placements while others are not visible on the first screen (Fig. 1). Another bias, selection bias, arises from imbalanced product impressions, where certain items—such as conversion offers—are shown to users more frequently, resulting in significantly higher exposure for those items.

3.2. Selection of Debiasing techniques

The primary reasoning for the selection of debiasing techniques for this study was based in a literature review, and included the applicability of each method to the specific biases present in the propreitery datasets—namely, selection bias, exposure bias, and position bias. Further, it was imperative to evaluate techniques across two dimensions: those that require randomized datasets and those that do not, as well as to examine methodologies that are agnostic to any particular type of bias. Given the identified biases in the datasets, we adopted several debiasing techniques: (1) **Matrix Factorisation** (MF) as a baseline model, **Inverse Propensity Scoring** (IPS), a method that does not require randomized data collection and primarily addresses selection and exposure biases. (2) **Doubly Robust learning**, that tackles the same biases but, unlike IPS, requires a randomized dataset. And (3) **AutoDebias** (DR), a bias-agnostic technique that also needs randomized data. Each method was tested across all datasets to evaluate model performance and complexity. We initially applied MF to biased dataset D_T to establish metrics for comparison, we denote our baseline model as **MF(biased)**, then compared these outcomes with the results from the debiasing methods.

3.3. Evaluation metrics

For models' evaluation, we use metrics that assess both predictive power of the models (RMSE and AUC), as well as quality of ranking (NDCG@5) and inequality and diversity in the recommendations (Gini index and Entropy):

- **NDCG@5** assesses the model's ability to rank relevant items in the recommendation list:

  $$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}, \quad \text{DCG@k} = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)},$$

  where IDCG@k is the ideal DCG@k and rel_i represents items ordered by their relevance up to position k.

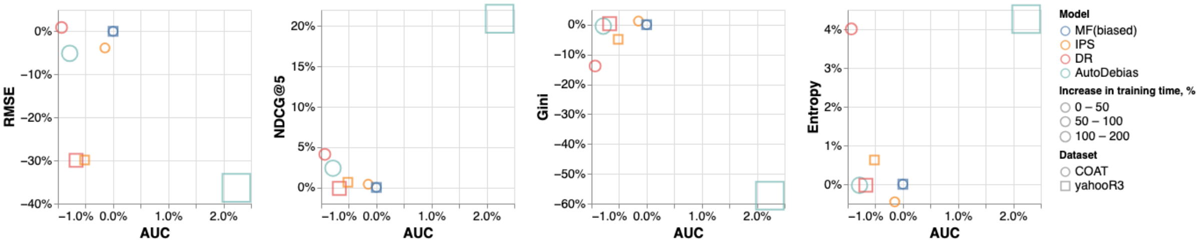

Figure 2: Debiasing results on open datasets (COAT and yahooR3!). The graphs show the percentage change in metrics (AUC, RMSE, NDCG@5, Gini, and Entropy) for various models relative to MF(biased). AUC is plotted against other metrics to demonstrate the trade-off between diversity gains in recommendation systems and potential compromises in predictive power. Different models are represented by colors, training times are indicated by point sizes, and dataset types are distinguished by shapes.

Table 2
Percentage improvement of various models compared to MF(biased) across open datasets. The best results for each metric are highlighted in bold.

Dataset	Model	RMSE	AUC	NDCG	Gini	Entropy	Training time (sec)
COAT	IPS	-2.53%	-0.26%	-1.18%	0.62%	-0.29%	**8.82%**
	DR	3.86%	-1.57%	2.75%	**-18.88%**	**6.16%**	194.12%
	AutoDebias	**-5.06%**	**0.39%**	**3.73%**	0.16%	0.00%	767.65%
yahooR3!	IPS	-29.70%	-0.55%	0.73%	-6.33%	0.82%	**-22.98%**
	DR	-30.39%	-0.83%	0.00%	1.22%	-0.12%	412.56%
	AutoDebias	**-36.89%**	**1.79%**	**20.70%**	**-58.15%**	**4.26%**	3215.87%

- **RMSE** measures the magnitude of prediction errors of exact rating predictions:

$$\text{RMSE} = \sqrt{\frac{1}{|R|}\sum_{(u,i)\in R}(\hat{r}_{ui} - r_{ui})^2},$$

  where $|R|$ denotes the total number of ratings in the dataset, $\hat{r}_{ui}$ and r_{ui} are predicted and true ratings for all user-item pairs (u, i).
- **AUC** reflects how well the model distinguishes between positive and negative interactions:

$$\text{AUC} = \frac{\sum_{(u,i)\in D^+_{\text{te}}} \text{rank}_{u,i} - \frac{(|D^+_{\text{te}}|+1)\cdot|D^+_{\text{te}}|}{2}}{|D^+_{\text{te}}|\cdot(|D_{\text{te}}| - |D^+_{\text{te}}|)},$$

  where D^+_{te} is the number of positive samples in test set D_{te}, and $rank_{u,i}$ denotes the position of a positive feedback (u, i). In experimentation, AUC mainly served as a metric to prevent overfitting and help fine-tunning in validation phase.
- **Gini index** measures inequality in the recommendations distribution. The higher coefficient indicates higher inequality

$$G = \frac{\sum_{i=1}^{n}(2i - n - 1)\,\phi_{(i)}}{n \cdot \sum_{i=1}^{n}\phi_{(i)}}$$

  Where ϕ_i is the popularity score of the i-th item, with the scores ϕ_i arranged in ascending order $(\phi_i \leq \phi_{i+1})$, and n represents the total number of items.
- **Entropy** measures the diversity in the distribution of recommended items with higher values indicating higher diversity.

$$Entropy = -\sum_{i=1}^{n} p_i \log(p_i),$$

  where n is a total number of items u in a dataset and p_i is a probability of an item being recommended.

Additionally, we include **Training Time**, defined as the time required for each model to reach saturation, measured in seconds. This metric provides insights into the computational complexity and the resources required by different methodologies.

4. Experimentation

We regard biased data as training set, D_T. When it comes to randomized data, following the strategies as mentioned in [11], we split it into 3 parts: 5% for randomised set D_U to help training as required by DR and Autodebias, 5% for validation set D_V to tune hyper-parameters and incur early-stopping mechanism to prevent overfitting, the rest 90% for test set D_{Te} to evaluate the model. For conformity reasons, the data split strategy mentioned above is applied to both open datasets and proprietary datasets. For this project, we deploy a training pipeline on Vertex AI [12], integrating components such as data transformation powered by BigQuery, model training and evaluation, as well as experiment tracking. The training pipeline retrieves data from the data warehouse to train models and produces artifacts that are later integrated into an experiment tracker. By adopting this artifact-based approach, we address the inherent challenge of reproducibility in operationalizing ML projects, as it provides all the necessary components to reproduce experiments. Each experiment is run up to 10 times on Vertex AI with the same hyper parameters, but varying random seeds to get estimation on the variability of the results.

A pipeline plays a pivotal role in enhancing machine learning processes within the industry by automating each step from data fetching to model evaluation. For this project, a training pipeline was implemented on Vertex AI, encompassing components such as data transformation utilizing BigQuery, model training, model evaluation, and experiment tracking. All the experiments were conducted within

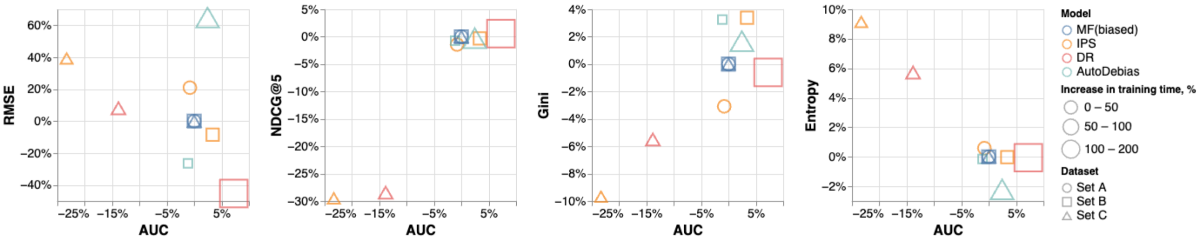

Figure 3: Debiasing results on internal datasets (Set A, Set B and Set C). The graphs show the percentage change in metrics (AUC, RMSE, NDCG@5, Gini, and Entropy) for various models relative to MF(biased). AUC is plotted against other metrics to demonstrate the trade-off between diversity gains in recommendation systems and potential compromises in predictive power. Different models are represented by colors, training times are indicated by point sizes, and dataset types are distinguished by shapes.

Table 3
Percentage improvement of various models compared to MF(biased) across internal datasets. The best results for each metric are highlighted in bold.

Dataset	Model	RMSE	AUC	NDCG	Gini	Entropy	Training time (sec)
Set A	IPS	20.95%	-0.97%	-1.53%	-3.06%	0.41%	-4.72%
Set B	IPS	-8.61%	3.18%	-0.14%	3.29%	-0.02%	**-12.23%**
	DR	**-45.40%**	**7.07%**	**0.68%**	**-0.54%**	**0.00%**	386.46%
	AutoDebias	-26.46%	-1.25%	-0.48%	3.26%	-0.02%	-63.26%
Set C	IPS	39.01%	-23.46%	-29.36%	**-9.47%**	**9.04%**	**-15.50%**
	DR	**7.74%**	-13.76%	-28.44%	-5.36%	5.47%	14.74%
	AutoDebias	64.50%	**2.61%**	**-0.01**%	1.72%	-2.47%	233.93%

this framework, ensuring consistency, efficiency, and precision throughout the development lifecycle.

5. Experimentation results

The absolute results of all experiments, including confidence intervals, are presented in Table 4. In this section, we report the percentage improvement of various debiasing techniques compared to the baseline model, which was trained on biased data (MF(biased) model).

5.1. Open Datasets

For the **COAT** dataset, the results show varying degrees of improvement across different metrics (Table 2). The top performing method (**AutoDebias**), exhibited the best improvements in RMSE (-5.06%), AUC (0.39%) and NGCG@5 (3.73%) with low changes in Gini (0.16%) and no improvement in Entropy. **DR** also provided higher gains in NDCG@5 (2.75%), and performed better in Gini (-18.88%) and Entropy (6.16%), but at a cost of higher RMSE (3.86%) and lower AUC (-1.57%). While AutoDebias outperformed other techniques when it comes to improving predictive power of the model (AUC, RMSE), it was not very efficient in terms of Gini and Entropy, and has a significantly higher computational cost. This highlights a trade-off between improved accuracy and increased resource requirements.

For **YahooR3!** dataset, again, **AutoDebias** results in the highest improvement in RMSE (-36.89%), AUC (1.79%), NDCG@5 (20.70%), as well as Gini (-58.15%) and Entropy (4.26%), but did so also with dramatically increased computational cost (3216%). **IPS** provides a balanced performance with improvements in RMSE (-29.70%) and Entropy (0.82%) at a lower computational cost (-22.98%), making it a practical choice for resource-constrained environments.

5.2. Internal Datasets

For the internal datasets, the results are less consistent across the datasets and debiasing techniques (Table 3). This may be due to the fact that internal datasets employed implicit feedback when collecting data, where user preferences are inferred from their impression and purchase records. This can introduce biases due to the lack of negative samples and overrepresentation of user interactions, potentially skewing the models towards popular items.

Set A is a relatively small dataset (Table 1), and the lack of randomized data limits our options to only using **IPS**. As a result, some metrics, such as RMSE and AUC, actually worsen (Table 3), which we might accept as a trade-off to achieve better balance in recommendations. However, NDCG@5 also does not improve. On the positive side, IPS enhances diversity metrics, with Gini improving by 3.06% and Entropy by 0.41%, while also reducing computational cost by 4.27%. Overall, applying this method increases model diversity with comparable training time, but comes at the cost of accuracy.

Set B demonstrates substantial improvements with **DR**, including a 45.40% reduction in RMSE, a 7.07% increase in AUC, and gains in NDCG@5 (0.68%) and Gini (-0.54%), making the model perform better in both accuracy and diversity. However, this comes at a significant computational cost, increasing training time by 386.46%. Given the total number of samples being 318k, this leads to a considerably longer training process. **AutoDebias** ranks second in RMSE improvement (-26.46%), while **IPS** shows a positive gain in AUC (3.18%). However, DR is the only method that consistently improves outcomes of NDCG@5, Gini, and Entropy.

For **Set C**, the largest dataset with nearly 2.2 million samples, **AutoDebias** achieves the highest improvement in AUC (2.61%) and maintains stable NDCG@5. However, it underperforms compared to the baseline and other tech-

Table 4
Performance metrics across different models and datasets, with 95% confidence intervals.

Dataset	Model	RMSE	AUC	NDCG@5	Gini	Entropy	Training time (sec)
COAT	MF (uniform)	1.00 ± 0.02	0.54 ± 0.01	0.36 ± 0.02	0.64 ± 0.01	4.91 ± 0.02	2.00 ± 1.60
	MF (biased)	0.75 ± 0.01	0.77 ± 0.01	0.51 ± 0.01	0.64 ± 0.04	4.9 ± 0.11	**3.40 ± 1.00**
	IPS	0.73 ± 0.01	0.76 ± 0.01	0.50 ± 0.01	0.65 ± 0.04	4.89 ± 0.10	3.70 ± 2.30
	DR	0.78 ± 0.02	0.75 ± 0.01	0.52 ± 0.01	**0.52 ± 0.01**	**5.20 ± 0.03**	10.00 ± 6.90
	AutoDebias	**0.71 ± 0.01**	**0.77 ± 0.02**	**0.53 ± 0.01**	0.64 ± 0.06	4.90 ± 0.14	29.50 ± 9.6
yahooR3!	MF (uniform)	0.73 ± 0.01	0.57 ± 0.01	0.43 ± 0.01	0.41 ± 0.01	6.58 ± 0.01	4.80 ± 1.20
	MF (biased)	0.86 ± 0.01	0.73 ± 0.01	0.55 ± 0.01	0.41 ± 0.01	6.58 ± 0.01	60.50 ± 12.20
	IPS	0.61 ± 0.01	0.72 ± 0.01	0.55 ± 0.01	0.39 ± 0.01	6.63 ± 0.02	**46.60 ± 16.10**
	DR	0.60 ± 0.04	0.72 ± 0.01	0.55 ± 0.01	0.42 ± 0.01	6.57 ± 0.01	310.10 ± 54.60
	AutoDebias	**0.54 ± 0.01**	**0.74 ± 0.01**	**0.66 ± 0.01**	**0.17 ± 0.01**	**6.86 ± 0.01**	2006.10 ± 1541.00
Set A	MF (biased)	**0.82 ± 0.07**	**0.54 ± 0.02**	**0.56 ± 0.02**	0.36 ± 0.01	2.83 ± 0.01	694.30 ± 163.30
	IPS	0.99 ± 0.02	0.54 ± 0.01	0.55 ± 0.01	**0.35 ± 0.02**	**2.84 ± 0.02**	661.5 ± 85.9
Set B	MF (uniform)	0.61 ± 0.00	0.92 ± 0.01	0.97 ± 0.00	0.10 ± 0.00	1.77 ± 0.00	2891.00 ± 126.90
	MF (biased)	0.81 ± 0.06	0.89 ± 0.00	0.97 ± 0.00	0.10 ± 0.00	1.80 ± 0.00	2123.90 ± 441.3
	IPS	0.74 ± 0.14	0.92 ± 0.01	0.97 ± 0.00	0.10 ± 0.00	1.77 ± 0.00	1864.10 ± 86.70
	DR	**0.44 ± 0.02**	**0.95 ± 0.01**	**0.96 ± 0.01**	**0.10 ± 0.01**	1.77 ± 0.00	10332.00 ± 2486.30
	AutoDebias	0.56 ± 0.02	0.88 ± 0.01	0.96 ± 0.01	0.10 ± 0.00	1.77 ± 0.00	**780.30 ± 153.70**
Set C	MF (uniform)	0.92 ± 0.04	0.25 ± 0.02	0.07 ± 0.01	0.52 ± 0.01	2.52 ± 0.02	775.90 ± 265.00
	MF (biased)	**0.62 ± 0.01**	0.84 ± 0.01	0.80 ± 0.01	0.65 ± 0.01	2.18 ± 0.02	650.80 ± 114.70
	IPS	0.86 ± 0.06	0.64 ± 0.05	0.56 ± 0.08	**0.59 ± 0.01**	**2.37 ± 0.02**	**549.90 ± 128.30**
	DR	0.67 ± 0.02	0.72 ± 0.05	0.57 ± 0.09	0.61 ± 0.02	2.29 ± 0.05	746.70 ± 140.00
	AutoDebias	1.02 ± 0.03	**0.86 ± 0.04**	**0.78 ± 0.02**	0.66 ± 0.02	2.12 ± 0.04	2173.20 ± 1826.10

niques in RMSE, Gini, Entropy, and training time, which increases significantly by 233.93%. **IPS**, on the other hand, delivers poor results in RMSE (39.01%), AUC (-23.46%), and NDCG@5 (-29.36%), but excels in Gini (-9.47%) and Entropy (9.04%) without adding to the training time.

6. Conclusion and Future work

Implementing more accurate and less biased models is crucial to avoiding the perpetuation of negative feedback loops and the overexposure of certain items caused by segmentation heuristics in retraining data. This approach also enhances data quality, which is essential for fine-tuning models. A recommender system that diversifies content exposure improves user experience by ensuring that visibility is not limited to only the most popular items. In our experiments, Inverse Propensity Scoring (IPS) stands out for its simplicity and model-agnostic nature, requiring no randomized data collection and fewer training epochs. However, the improvements it offers are somewhat limited. AutoDebias excels in improving accuracy metrics, but at substantially higher computational costs and sometimes poorer performance in Gini and Entropy. DR still offers strong improvement in observed metrics, including Gini and Entropy. So while each debiasing method has its own trade-offs, significant performance gains still depend on the challenging task of collecting randomized datasets, as highlighted in our introduction. Potential future work includes: (1) adopting online reinforcement learning approach such as Multi-Armed Bandit (MAB) [14, 15, 16] for data collection, including contextual bandit models, (2) developing and testing combined debiasing models which can combine strengths of different debiasing techniques to mitigate various biases simultaneously while optimizing for computational efficiency.

References

[1] Oard, Douglas W.& Jinmook Kim. *Implicit feedback for recommender systems.* Proceedings of the AAAI workshop on recommender systems. Vol. 83. 1998.

[2] Jiawei Chen and Hande Dong and Xiang Wang and Fuli Feng and Meng Wang & Xiangnan He, *Bias and Debias in Recommender System: A Survey and Future Directions.* ACM Trans. Inf. Syst. 41, 3, Article 67 (2023)

[3] Harald Steck, *Training and testing of recommender systems on data missing not at random.* In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2010). Association for Computing Machinery, New York, NY, USA, 713–722

[4] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims, *Recommendations as Treatments: Debiasing Learning and Evaluation.* In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML 2016). JMLR.org, 1670–1679.

[5] Quanyu Dai, Haoxuan Li, Peng Wu, Zhenhua Dong, Xiao-Hua Zhou, Rui Zhang, Rui Zhang, and Jie Sun, *A Generalized Doubly Robust Learning Framework for Debiasing Post-Click Conversion Rate Prediction.* In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022). Association for Computing Machinery, New York, NY, USA, 252–262.

[6] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang, *AutoDebias: Learning to Debias for Recommendation.* In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). Association for Computing Machinery, New York, NY, USA, 21–30.

[7] Andrés Villa, Vladimir Araujo, Francisca Cattan, and

Denis Parra, *Interpretable Contextual Team-aware Item Recommendation: Application in Multiplayer Online Battle Arena Games.* In Proceedings of the 14th ACM Conference on Recommender Systems (RecSys 2020)

[8] Qilin Deng, Kai Wang, Minghao Zhao, Zhene Zou, Runze Wu, Jianrong Tao, Changjie Fan, and Liang Chen *Personalized Bundle Recommendation in Online Games.* In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM 2020)

[9] Meng Wu, John Kolen, Navid Aghdaie, and Kazi A. Zaman. *Recommendation Applications and Systems at Electronic Arts.* In Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys 2017)

[10] Cao, Lele & Asadi, Sahar & Biasielli, Matteo & Sjöberg, Michael. *Debiasing Few-Shot Recommendation in Mobile Games.* Workshop of ACM Conference on Recommender Systems (RecSys 2020)

[11] Liu, Dugang and Cheng, Pengxiang and Dong, Zhenhua and He, Xiuqiang and Pan, Weike and Ming, Zhong *A general knowledge distillation framework for counterfactual recommendation via uniform data.* Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)

[12] Google. 2024, *Vertex AI.* Retrieved December 1, 2023 from https://cloud.google.com/vertex-ai,

[13] Marlin, Benjamin M., and Richard S. Zemel. *Collaborative prediction and ranking with non-random missing data.* Proceedings of the third ACM conference on Recommender systems (RecSys 2009)

[14] Felício, Crícia Z., Klérisson VR Paixão, Celia AZ Barcelos, and Philippe Preux. *A multi-armed bandit model selection for cold-start user recommendation.* In Proceedings of the 25th conference on user modeling, adaptation and personalization, pp. 32-40. 2017.

[15] Wang, Lu, Chengyu Wang, Keqiang Wang, and Xiaofeng He. *Biucb: A contextual bandit algorithm for cold-start and diversified recommendation.* In 2017 IEEE International Conference on Big Knowledge (ICBK), pp. 248-253. IEEE, 2017.

[16] Wang, Qing, Chunqiu Zeng, Wubai Zhou, Tao Li, S. Sitharama Iyengar, Larisa Shwartz, and Genady Ya Grabarnik. *Online interactive collaborative filtering using multi-armed bandit with dependent arms.* IEEE Transactions on Knowledge and Data Engineering 31, no. 8 (2018): 1569-1580.