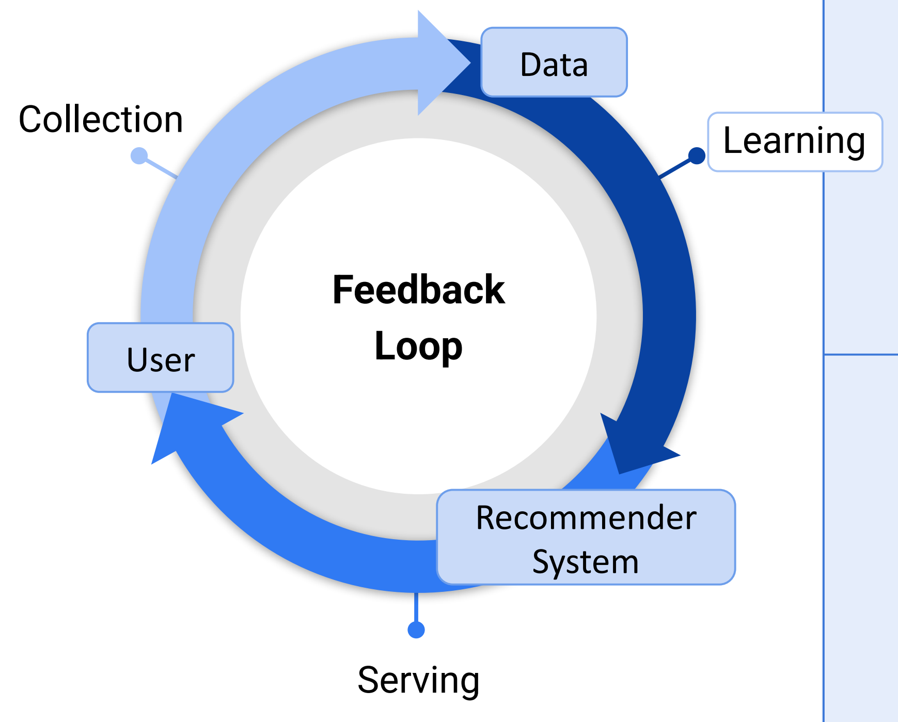
# The Effects of Debiasing Methods on the Fairness and Accuracy of Recommender Systems

Author: Filip Čajági (f.cajagi@student.tudelft.nl)     Supervisor: Masoud Mansoury (m.mansoury@tudelft.nl)

## 1. Introduction

**Recommender systems** have become an integral part of most people's everyday lives: video/movie recommendations, social media posts, e-shop listings, dating apps and more. They leverage users' past interactions to predict preferences and deliver personalized recommendations to improve the user experience.

However, due to how these systems collect and learn from data, they often suffer from various biases, with one of the most pervasive being the **popularity bias** – when few popular items get over-recommended, while most other items get under-recommended. Additionally, there is also the question of **fairness** - whether different user and item groups receive recommendations of similar quality, and whether all items are exposed equitably.



Researchers in the past have tried solving the bias problem with various **debiasing methods** including pre-processing data augmentation, post-processing re-ranking as well as in-processing methods. Fairness, on the other hand, is often studied in a separate strand of research with **fairness intervention methods**. As a result the effects of debiasing methods on the fairness metrics remains an underexplored area, and the trade-off between fairness and accuracy is rarely explicitly evaluated, which this project aims to address.

## 2. Research Questions

Main research question:

- Do **debiasing methods** contribute to mitigating the **fairness issue**, or do they primarily improve **accuracy** without addressing fairness directly?

Sub-questions:

- **RQ1:** How do debiasing methods **affect the trade-off** between fairness and accuracy in recommender systems?
- **RQ2:** Can **varying** the **hyper-parameters** of debiasing methods be used to **control** their effect on the performance of recommender systems and the trade-off between fairness and accuracy in recommender systems?

## 3. Key Definitions

- **Popularity bias:** Few items get over-recommended, more than their relevance would warrant, while most other items suffer from under-exposure. Propagates itself in many recommender systems through the feedback loop.
- **Selection bias:** Users choose which items they want to rate, instead of rating all items they interacted with, which results in the data not being fully representative of the users' real preferences.
- **Item-side fairness:** Whether all items (e.g. products, movies, or other users in some systems) receive equitable exposure to users.
- **User-side fairness:** Whether all users or user groups (e.g. gender groups or minorities) receive recommendations of similar quality (i.e. relevant to their interests).

## 4. Methodology

### 4.1 Algorithms

The recommender system algorithms/models chosen for the experiments are:

**Baselines:**

- **Random** – Recommends random items to all users
- **Pop** – Recommends the most popular items to all users
- **ItemKNN/UserKNN** – A traditional memory-based k-nearest-neighbors approach using the similarity of users/items based on their interactions
- **MF** (Matrix Factorization) – A more nuanced approach that estimates lower-dimensional embeddings for users and items based on their interactions

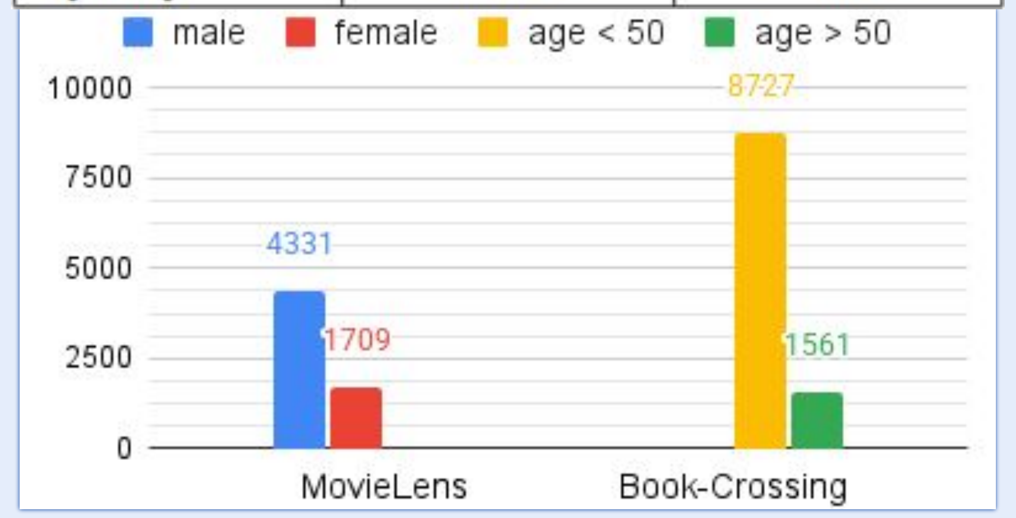
**Debiasing Methods** applied on the MF model:

- **IPS** (Inverse propensity scoring) – Weighs the effect of interactions on the training process based on the item's popularity, with unlikely interactions having greater effect. (Schnabel et al., 2016)
- **PDA** (Popularity-bias Deconfounding and Adjusting) – Decouples the effect of item popularity during training and then injects it back into the recommendations to eliminate the amplification of the bias, while still leveraging the popularity for better recommendations (Zhang et al., 2021)
- **MACR** (Model-Agnostic Counterfactual Reasoning) – Simultaneously trains a regular model, and two supporting modules based solely on the item and user embeddings, and then uses counterfactual inference to subtract the supporting modules from the main one. (Wei et al., 2021)

### 4.2 Datasets

The datasets chosen for the experiments are: MovieLens 1M and Book-Crossing. After retaining only users and items with at least 5 interactions (5-core filtering), and only users with a valid age, these are the statistics of the datasets:

| Property | MovieLens 1M | Book-Crossing |
|---|---|---|
| Users | 6,040 | 10,288 |
| Items | 3,416 | 28,171 |
| U-I interactions | 999,611 | 399,240 |
| Sparsity | 95.157% | 99.862% |



### 4.3 Metrics

The results will be evaluated in terms of the following accuracy and fairness metrics (@K means the metric is evaluated on a list of top-K recommendations):

| Accuracy: | Item-side fairness: | User-side fairness: |
|---|---|---|
| Recall@K | **ItemCoverage@K** | NonParity |
| Precision@K | TailPercentage@K | nDCG_Maj@K |
| **nDCG@K** | PopularPercentage@K | nDCG_Min@K |
| Hit@K | GiniIndex@K | **nDCG_Diff@K** |

The full results are available in the paper, but only one main metric from each category was chosen for this poster:

- **nDCG@K** (normalized discounted cumulative gain) - Measure of ranking quality that assigns higher scores to hits at top ranks
- **ItemCoverage@K** - Coverage of all recommended items over all items
- **nDCG_Diff@K** - Disparity of the nDCG metric between the majority and minority groups, computed as (1 - nDCG_Min / nDCG_Maj)

## 5. Results

### 5.1 RQ1: Fairness and Accuracy trade-off of Debiasing Methods

The following table shows the results of the chosen 3 metrics for all the models, using default hyper-parameter values, on both the datasets:

| Method | MovieLens 1M | | | Book-Crossing | | |
|---|---|---|---|---|---|---|
| | nDCG@10 | ItemCov@10 | nDCG_Diff@10 | nDCG@10 | ItemCov@10 | nDCG_Diff@10 |
| ItemKNN | 0.2549 | 0.3878 | 0.2124 | 0.0486 | 0.7096 | 0.3006 |
| UserKNN | **0.2682** | 0.2075 | 0.2134 | **0.0493** | 0.2024 | 0.1680 |
| Random | 0.0061 | **0.9997** | 0.2424 | 0.0003 | **0.9743** | *-3.5000* |
| Pop | 0.1218 | 0.0293 | 0.3390 | 0.0161 | 0.0012 | **0.1515** |
| MF | 0.2520 | 0.4287 | **0.1964** | 0.0230 | 0.1111 | 0.2583 |
| MF-IPS | 0.1986 | **0.6043** | 0.2071 | 0.0190 | 0.1116 | **0.0521** |
| MF-PDA | **0.2673** | 0.2426 | **0.1911** | 0.0200 | 0.0137 | 0.2270 |
| MF-MACR | 0.2466 | 0.5505 | 0.2080 | **0.0226** | **0.2502** | 0.0746 |

***Discussion***

While traditional neighborhood-based models perform the best in terms of accuracy, they are not particularly efficient, especially for larger datasets.

As for the debiasing methods, we can see the fairness and accuracy trade-off, with PDA increasing nDCG while decreasing ItemCov. On the other hand, IPS and MACR increase the coverage at the cost of accuracy. The most favorable trade-off can be seen using **MACR** on the Book-Crossing dataset, which **increases item coverage by 225% with only a minimal, 2% cost in nDCG,** while also notably decreasing nDCG_Diff by increasing the recommendation quality for the minority group.

***RQ1 Answer:***

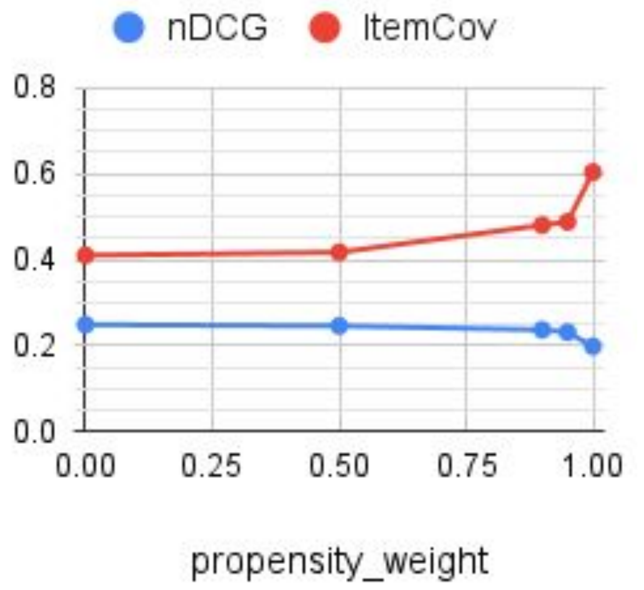
Overall, debiasing methods can have a **significant impact on the trade-off** between fairness and accuracy of recommender models, but the specifics **heavily depend on the method** as well as the dataset used. Both accuracy and fairness can be increased, usually at the cost of the other, although in some cases both can also be increased or decreased together. Of the three debiasing methods evaluated, **MACR performed the best** in terms of the trade-off, with minimal impacts to accuracy and significant improvements in both item-side and user-side fairness.

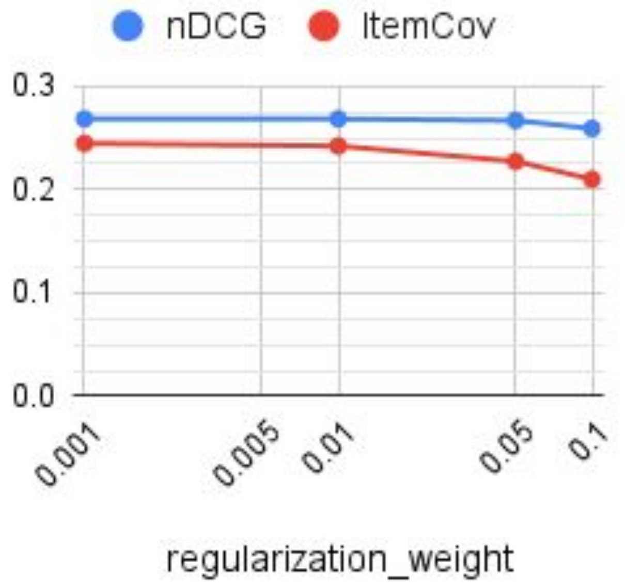
### 5.2 RQ2: Hyper-parameters of Debiasing Methods

Since each debiasing method has its own different hyper-parameters, we will analyze them separately on the MovieLens dataset.

***IPS***

- propensity_weight controls the impact of the debiasing method.
- Increasing it improves coverage, but lowers accuracy, allowing us to control the trade-off as we need to.
- Most significant effect at values near 1
- User-side fairness was unaffected in this case.



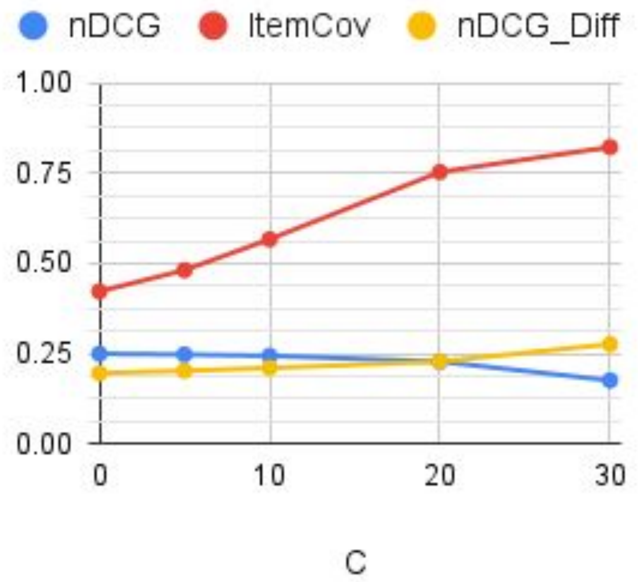


***PDA***

- Applies L2 regularization on the user and item embeddings, which can be controlled with regularization_weight.
- In our case the default value of 0.001 has almost no effect on the model and increasing it is only detrimental.
- While this regularization was not useful in our experiments, it might be beneficial for larger models and datasets.

***MACR***

- The C parameter scales the impact of the user and item submodules.
- Shows the greatest flexibility in terms of controlling the trade-off.
- Greatly improves item coverage with minimal impacts to accuracy.
- Higher values have diminishing results.
- User-side fairness was also slightly negatively affected.



***RQ2 Answer:***

The **hyper-parameters** of debiasing methods can be used to **control the trade-off** between fairness and accuracy based on the needs of the developer, with some methods being more flexible than others. Out of the three analyzed debiasing methods, **MACR shows the greatest flexibility** in terms of controlling the trade-off and increasing item-side fairness.

## 6. Conclusion

- Overall, we found that debiasing methods can have a **significant impact on the trade-off** between fairness and accuracy of recommender models, and **hyper-parameters can be used to control it**, although the specifics heavily depend on the used method as well as dataset.

***Limitations and Future work:***

- The scope of our project was somewhat limited - only 3 debiasing methods, 2 datasets and 1 base model, so future work in the area can focus on **expanding the scope**
- Our metrics, or similar ones, can be used to monitor the performance and trade-offs of any debiasing method or recommender model.