Multi-sided Exposure Bias in Recommendation

Himan Abdollahpouri
himan.abdollahpouri@colorado.edu
University of Colorado Boulder
Boulder, USA

Masoud Mansoury*
m.mansoury@tue.nl
Eindhoven University of Technology
Eindhoven, The Netherlands

ABSTRACT

Academic research in recommender systems has been greatly focusing on the accuracy-related measures of recommendations. Even when non-accuracy measures such as popularity bias, diversity, and novelty are studied, it is often solely from the users' perspective. However, many real-world recommenders are often multi-stakeholder environments in which the needs and interests of several stakeholders should be addressed in the recommendation process. In this paper, we focus on the popularity bias problem which is a well-known property of many recommendation algorithms where few popular items are over-recommended while the majority of other items do not get proportional attention and address its impact on different stakeholders. Using several recommendation algorithms and two publicly available datasets in music and movie domains, we empirically show the inherent popularity bias of the algorithms and how this bias impacts different stakeholders such as users and suppliers of the items. We also propose metrics to measure the exposure bias of recommendation algorithms from the perspective of different stakeholders.

KEYWORDS

Multi-sided platforms, Recommender systems, Popularity bias, Multi-stakeholder recommendation

1 INTRODUCTION

Popularity bias is a well-known phenomenon in recommender systems: popular items are recommended even more frequently than their popularity would warrant, amplifying the long-tail effect already present in many recommendation domains. Prior research has examined the impact of this bias on some properties of the recommenders such as aggregate diversity (aka catalog coverage) [4, 22]. One of the consequences of the popularity bias is disfavoring less popular items where the recommendations are not fair in terms of the amount of exposure they give to different items with varying degree of popularity: an exposure bias. However, as we discuss in [1], many recommender systems are multi-stakeholder environments in which the needs and interests of multiple stakeholders should be taken into account in the implementation and evaluation of such systems.

In many multi-stakeholer recommenders as described in [1] two main stakeholders (or what often is being referred to as *sides* in multi-sided platforms [11]) can be identified: consumers (aka users) and suppliers. For instance, in a music platform such as Spotify, on one side there are users who get recommendations for songs in which they are interested and, on the other side, there are artists whose songs are being recommended to different users. The popularity bias can be investigated from both sides' perspective.

Regarding the users, not everyone has the same level of interest in popular items. In the music domain as an example, some users might be interested in internationally popular artists such as Drake, Beyoncé, or Ed Sheeran and some might be more interested in artists from their own culture that might not necessarily have the same popularity as the aforementioned artists (such as the Iranian musician Kayhan Kalhor) or generally they prefer certain type of music that might not be popular among the majority of other users (such as country music). With that being said, we expect the personalization to handle this difference in taste but as we will see in section 4.1 that is certainly not the case.

The suppliers also do not have the same level of popularity. In many recommendation domains including movies, music, or even house sharing, few suppliers have a large audience while the majority of others may not be as popular though they still might have their fair share of audience. Now the question is, do recommender systems let different suppliers with varying degree of popularity reach their desired audience? Again, the short answer is No as we will see more details in section 4.2.

Investigating the impact of recommendation algorithms on the exposure bias on both users and suppliers is the focus of this paper. We study several recommendation models in terms of their inherent popularity bias and propose metrics that can measure such impact.

2 EXPERIMENTAL SETTING

2.1 Data

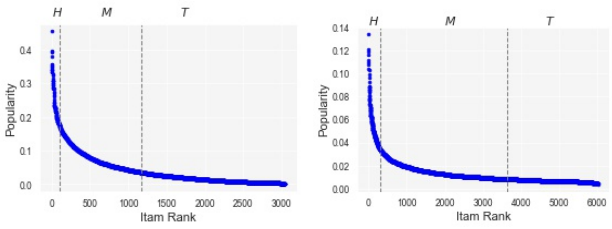
We have used two publicly available datasets for our experiments. We needed datasets that either had information about the supplier of the items or we could extract them. We found two: the first one is a sample of the Last.fm (LFM-1b) dataset [19] used in [9]. The dataset contains user interactions with songs (and the corresponding albums). We used the same methodology in [9] to turn the interaction data into rating data using the frequency of the interactions with each item (more interactions with an items will result in higher ratings). In addition, we used albums as the items to reduce the size and sparsity of the item dimension, therefore the recommendation task is to recommend albums to users. We considered the artists of each album as the supplier. Each album is associated with an artist. Artists could have multiple albums. We removed users with less than 20 ratings so only consider users for which we have enough data. The resulting dataset contains 274,707 ratings by 2,697 users to 6,006 albums from 1,998 artists.

The second dataset is the MovieLens 1M dataset[1]. This dataset does not have the information about the suppliers. We considered the director of each movie as the supplier of that movie and we extracted that information from the IMDB API. Total number of ratings in the MovieLens 1M data is 1,000,209 given by 6,040 users to 3,706 items. Overall, we were able to extract the director information for 3,043 movies reducing the ratings to 995,487. The total number of directors is 831.

We used 80% of each dataset as our training set and the other 20% for the test.

*This author also has affiliation in School of Computing, DePaul University, Chicago, USA, mmansou4@depaul.edu.

[1]Our experiments showed similar results on MovieLens 20M, and so we continue to use ML1M for efficiency reasons.



(a) MovieLens (b) Last.fm

Figure 1: The Long-tail Distribution of Rating data. items are a small number of very popular items that take up around 20% of the entire ratings.) is the larger number of less popular items which collectively take up roughly 20% of the ratings, and " includes those items in between that receive around 60% of the ratings, collectively.

2.2 Algorithms

For our analysis, we have used three personalized recommendation algorithms: *Biased Matrix Factorization* (`Biased-MF`) [14], *User-based Collaborative Filtering* (`User-CF`) [5], and *Item-based Collaborative Filtering* (`Item-CF`) [8]. All algorithms are tuned to achieve their best performance in terms of precision. The size of the recommendation lists for each user is set to 10. We also include a non-personalized `Most-popular` algorithm which only recommends the 10 most popular items to every user given they have not rated the items before. We used LibRec [10] and *librec-auto* [16] for running the algorithms.

3 POPULARITY BIAS

Skew in wealth distribution is well-known: The richest 10% of adults in the world own 85% of global household wealth while the bottom half collectively owns barely 1% [2]; in recommender systems a similar problem exists: a small number of popular items appear frequently in user profiles and a much larger number of less popular items appear rarely. This bias can originate from two different sources: data and algorithms.

3.1 Bias in Data

Rating data is generally skewed towards more popular items. Figure 1 shows the percentage of users who rated different items in MovieLens and Last.fm datasets: the popularity of each item. Items are ranked from the most popular to the least with the most popular item being on the far left on the x-axis. Three different groups of items can be seen in these plots: which represents few items that are very popular and take up around 20% of the entire ratings according to the Pareto Principle [18].) is the larger number of less popular items which collectively take up roughly 20% of the ratings, and " includes those items in between that receive around 60% of the ratings, collectively. The curve has a long-tail shape [6, 7] indicating few popular items are taking up the majority of the ratings while many other items on the far right of the curve have not received much attention. This type of distribution can be found in many other domains such as e-commerce where few products are best-sellers, online dating where few profiles attract the majority of the attention, social networking platforms where few users have millions of followers, to name a few.

[2]https://www.wider.unu.edu/publication/global-distribution-household-wealth

The bias in rating data could be due to two different reasons:

External Bias: Some items and products are inherently more popular than others even outside of the recommender systems and in the real world. For instance, even before the music streaming services emerge, there were always few artists that were nationally or internationally popular such as *Shakira*, *Jennifer Lopez*, or *Enrique Iglesias*. As a result of this external bias (or tendency) towards popular artists, users also often listen to those artists more on streaming services and hence they get more user interactions.

Feedback Loop: Since the recommendation algorithms have a higher tendency towards recommending popular items, these items have a higher chance to be recommended to the users and hence garnering a larger number of interactions from the users. When these interactions are logged and stored, the popularity of those items in the rating data increases since they get more and more interactions over time [12, 20].

3.2 Bias in Algorithm

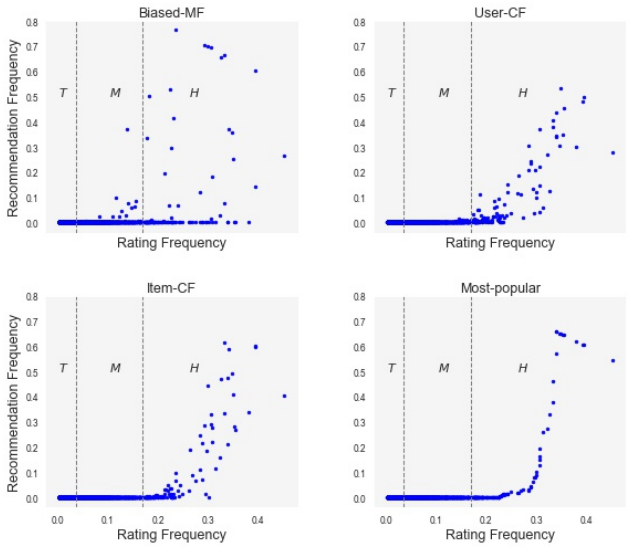
Due to this imbalance property of the rating data, often algorithms inherit this bias and, in many cases, intensify it by over-recommending the popular items and, therefore, giving them a higher opportunity of being rated by more users: *the rich get richer and the poor get poorer* [2].

Figure 2 shows the percentage of users who have rated an item on the x-axis (the popularity of an item in the data) and the percentage of users who received that item in their recommendations using four different recommendation algorithms `Biased-MF`, `User-CF`, `Item-CF`, and `Most-popular` in both datasets. The plots aim at showing the correlation between the popularity of an item in the rating data versus how often it is recommended to different users. It is clear that in all four algorithms, many items are either never recommended or just rarely recommended. Among the three personalized algorithms, `Item-CF` and `User-CF` show the strongest evidence that popular items are recommended much more frequently than the others. In fact, they are recommended to a much greater degree than even what their initial popularity warrants. For instance, the popularity of some items have been amplified from roughly 0.4 to 0.7 indicating a 75% increase. Both `Item-CF` and `User-CF` are over-promoting popular items (items on the right side of the x-axis) while significantly hurting other items by not recommending them proportionate to what their popularity in data warrants. In fact, the vast majority of the items on the left are never recommended indicating an extreme bias of the algorithms towards popular items and against less popular ones. `Biased-MF` does not show a positive correlation between popularity in data and in recommendations although some items are still over-recommended (have much higher popularity in recommendations versus what they had in rating data). However, this over-recommendation is not concentrated on only popular items and some items from lower popularity values are also over-recommended. `Most-popular` obviously shows the strongest bias but, unlike the other three, it is not a personalized method[3].

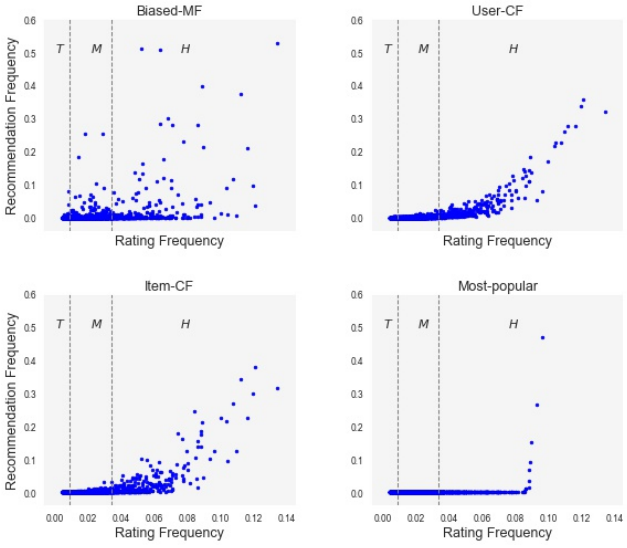
4 MULTI-SIDED EXPOSURE BIAS

We measure the impact of popularity bias on different stakeholders in terms of exposure: how the bias in algorithms prevents users

[3]If an item in the top 10 is already rated by a user it will be replaced by another popular item. That is why there is an inflection point in the scatter plot for this algorithm on MovieLens dataset.



(a) MovieLens



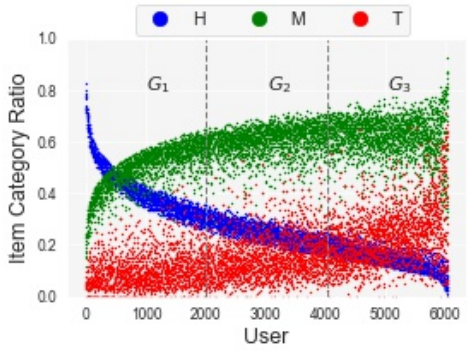
(b) Last.fm

Figure 2: Item popularity versus recommendation popularity

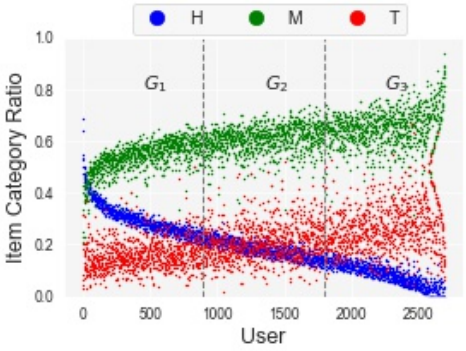
to be exposed to the appropriate range of items and also how it stops items from different suppliers to be exposed to their desired audience.

4.1 Exposure Bias From the Users' Perspective

Not every user is equally interested in popular items. In cinema, for instance, some might be interested in movies from Yasujiro Ozu, Abbas Kiarostami, or John Cassavetes, and others may enjoy more mainstream directors such as James Cameron or Steven Spielberg. Figure 3 shows the ratio of rated items for three item categories , ", and) in the profiles of different users in the MovieLens 1M and Last.fm datasets. Users are sorted from the highest interest



(a) MovieLens



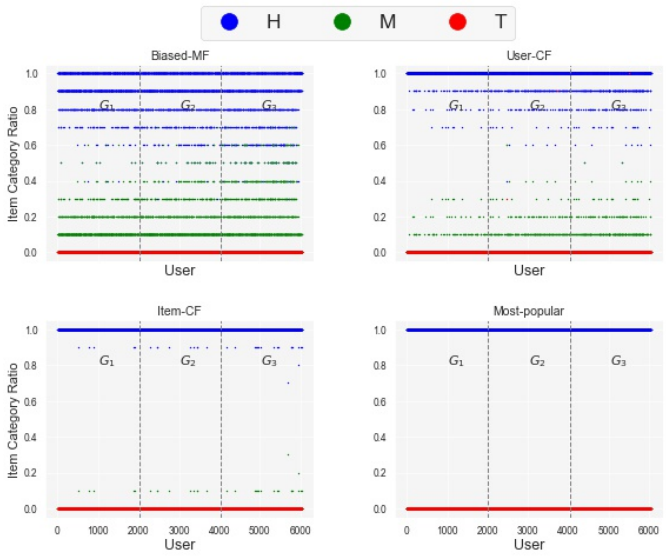
(b) Last.fm

Figure 3: Users' Propensity towards item popularity

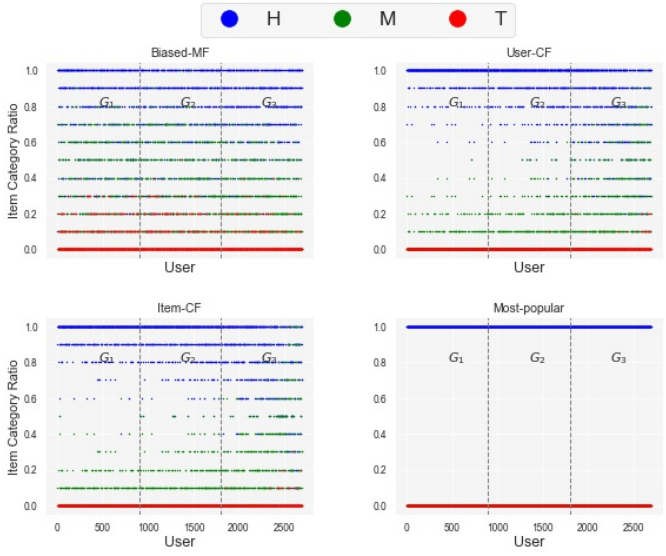
towards popular items to the least and divided into three equal-sized bins = f 1, 2, 3g from most popularity-focused to least. The y-axis shows the proportion of each user's profile devoted to different popularity categories. The narrow blue band shows the proportion of each users profile that consists of popular items (i.e. the category), and its monotone decrease reflects the way the users are ranked. Note, however, that all groups have rated many items from the middle (green) and tail (red) parts of the distribution.

The plots in Figure 4 are parallel to Figure 3, with the users ordered by their popularity interest, but now the y-axis shows the proportion of recommended items using different algorithms from different item popularity categories. The difference with the original user profiles in rating data especially in the case of `Most-popular`, `Item-CF`, and `User-CF` is stark where the users' profiles are rich in diverse popularity categories, the generated recommendations are nowhere close to what the user has shown interest at. In fact, in `Item-CF` almost 100% of the recommendations are from the head category, even for the users with the most niche-oriented profiles. Tail items do not appear at all. We demonstrated here that popularity bias in the algorithm is not just a problem from a global, system, perspective. It is also a problem from the user perspective [3]: users are not getting recommendations that reflect the diversity of their profiles and the users with the most niche tastes (3) are the most poorly served.

To measure the impact of popularity bias on users, we need to compare two lists together: the list of the items in a users' profile and the list of the items recommended to the user. To do this comparison,



(a) MovieLens



(b) Last.fm

Figure 4: Users' centric view of popularity bias

we need to compute a discrete probability distribution % for each user D, reflecting the popularity of the items found in their profile d_D over each item category 2 2 (in this paper, = f • "•) g). We also need a corresponding distribution & over the recommendation list given to user D, $_D$, indicating what popularity categories are found among the list of recommended items. In Steck [21] and Kaya et al. [13] where, unlike this paper, the calibration is done according to genres, the rating data is binarized to reflect just "liked" items. We are retaining the original ratings when computing over user profiles and, instead, using Vargas et al.'s [23] measure of category propensity which has the users' rating component in it. Note that unlike the genre labels in [21] where it is possible for a movie to have multiple genres, each item only has one popularity category.

$$?^{1}2jD^{o} = \frac{\acute{I}_{\delta 2 d_D A^{1}D \bullet \delta^{o}\ ^{1}\delta\ 2\ 2^{o}}}{\acute{I}_{2_{9}2\ \delta 2 d_D A^{1}D \bullet \delta^{o}\ ^{1}\delta\ 2\ 2_{9}{}^{o}}} \quad (1)$$

$$@^{1}2jD^{o} = \frac{\acute{I}_{\delta 2_D\ ^{1}\delta\ 2\ 2^{o}}}{\acute{I}_{2_{9}2\ \delta 2_D\ ^{1}\delta\ 2\ 2_{9}{}^{o}}} \quad (2)$$

$^{1}{}^{"o}$ is the indicator function returning zero when its argument is False and 1 otherwise. Once we have % and &, we can measure the distance using Jensen–Shannon divergence, which is a modification of KL-Divergence that has two useful properties which KL-divergence lacks: 1) it is symmetric: $J^{1}\%\bullet\&^{o} = J^{1}\&\bullet\%^{o}$ and 2) it has always a finite value even when there is a zero in &. For our application, it is particularly important that the function be well-behaved at the zero point since it is possible for certain item categories to have zero items in them in the recommendation list.

$$J^{1}\%\bullet\&^{o} = \frac{1}{2}\ !^{1}\%\bullet\ "^{o}\ ,\ \frac{1}{2}\ !^{1}\&\bullet\ "^{o}\bullet\quad " = \frac{1}{2}{}^{1}\%\ ,\ \&^{o} \quad (3)$$

where ! is the KL divergence.

We introduce the following metric to quantify the exposure bias from the users' perspective.

Users' Popularity propensity Deviation (UPD):

Having different groups of users with varying degree of interest towards popular items, we measure the average deviation of the recommendations given to the users in each group in terms of item popularity. More formally,

$$*\% = \frac{\acute{I}_{g2}\ \frac{\acute{I}_{D2g}\ J^{1}\%\ ^{1}d_D{}^{o}\bullet\&^{1}{}_{D}{}^{oo}}{\text{jgj}}}{\text{j}\quad\text{j}} \quad (4)$$

where jgj is the number of users in group g and j j is the number of user groups. * % can be also seen as the average miscalibration of the recommendations from the perspective of users in different groups.

4.2 Exposure Bias From the Suppliers' Perspective

As noted above, multi-stakeholder analysis in recommendation also includes providers or as termed here *suppliers*, "those entities that supply or otherwise stand behind the recommended items" [1]. We can think of many different kinds of contributors standing behind a particular movie: for the purposes of this paper, we will focus on movie directors. In the music domain, often the artists are considered as the suppliers of the songs [17]. In this paper, we also make the same assumption on Last.fm dataset.

We create three supplier groups (= f($_1$•($_2$•($_3$g: ($_1$ represents few popular suppliers whose items take up 20% of the ratings, ($_2$ are larger number of suppliers with medium popularity whose items take up around 60% of the ratings, and ($_3$ are the less popular suppliers whose items get 20% of the ratings. Figure 5 shows the rank of different directors in MovieLens and artists in Lats.fm datasets by popularity and the corresponding recommendation results from different algorithms. The recommendations have amplified the popularity of the popular suppliers (the ones on the extreme left) while suppressing the less popular ones dramatically. Strikingly, using `Item-CF`, movies from just 3 directors in ($_1$ (less than 0.4% of the suppliers here) take up 50% of the recommendations produced, while items items from the ($_3$ are seeing essentially zero recommendations.

To quantify the impact of popularity bias on the supplier exposure, we measure the amount of deviation different groups of suppliers experience in terms of exposure. In other words, the degree of over-recommendation or under-recommendation of suppliers from different groups.

Supplier Popularity Deviation (SPD):

$$(\% = \frac{\text{Í } B2(\text{ j@}^1B^o \quad ?^1B^o\text{j}}{\text{j(j}} \tag{5}$$

$$@^1B^o = \frac{\text{Í} \quad \text{Í}}{= \quad \text{j*j}} \frac{D2* \quad 82_D \quad ^1 \quad 1^o \; 2 \; B^o}{}, \; ?^1B^o = \frac{\text{Í} \quad \text{Í}}{\text{Í}} \frac{D2* \quad 82d_D \quad ^1 \quad 1^o \; 2 \; B^o}{D2* \; \text{j}d_D\text{j}}$$

where $@^1B^o$ is the ratio of recommendations that come from items of supplier group B. $?^1B^o$ is the ratio of ratings that come from items of supplier group B. $*$ is the set of users and $^1{}^n{}^o$ is a mapping function that returns the supplier for each item.

In fact, 1 $(\%$ can be considered as *Proportional Supplier Fairness* [15] since it measures how the items from different supplier groups are exposed to different users *proportional* to their popularity in rating data.

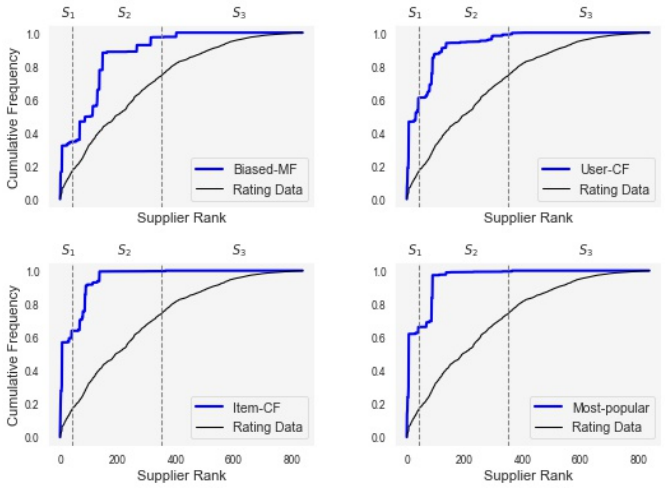
5 DISCUSSION AND FUTURE WORK

One important consideration regarding the deviation of popularity from the users' perspective ($*\%$) and suppliers' perspective ($(\%$) is how these two metrics behave with respect to one another. In other words, whether calibrating the recommendations for the users in terms of popularity would make the experience for the suppliers also better. Figures 6(a) and 7(a) show the connection between these two in the MovieLens and Last.fm datasets, respectively. We can see that, generally, the lower the $*\%$ (better calibration from the users' perspective) the lower the $(\%$ (better proportional fairness for the suppliers) will be. This indicates the advantage of optimizing for the popularity calibration in the recommendations since it will also benefit the suppliers.

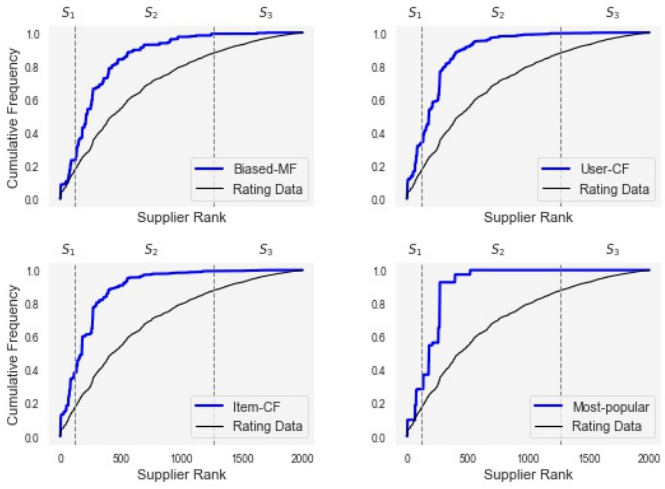
Another important finding is the fact that more accuracy does not necessarily leads to a better calibration and vice versa as can be seen from Figures 6(b) and 7(b). For instance, on Last.fm, `Most-popular` and `Biased-MF` have roughly equal precision but the $*\%$ for `Biased-MF` is significantly lower (better) than `Most-popular`. The reason is, the majority of the items are usually not very popular and, therefore, exclusively recommending popular items would not match the original distribution of the ratings (Figure 1) which leads to high $*\%$. In addition, if an algorithm randomly recommends items, the likelihood of having items from $''$ and $)$ increases compared to many personalized recommendations where they suffer from popularity bias. However, improvement in $*\%$ by randomly recommending items would happen under the cost of having an extremely low precision. Therefore, in practice, both $*\%$ and $\%A428B8>=$ should be taken into account simultaneously in the optimization process.

The relationship between catalog coverage and supplier popularity deviation is also interesting to look at. Will an algorithm that covers many items necessarily have lower (better) $(\%$? The answer is No as can be seen from Figures 6(c) and 7(c). The reason is, item coverage (aka aggregate diversity) is the number of unique items that an algorithm has recommended even if an item is only recommended only once. However, what matters in $(\%$ is giving appropriate exposure to the items from different suppliers proportional to their popularity.

The impact of $*\%$ and $(\%$ on the success of a real-world recommender system can be evaluated using online A/B testing and see how metrics such as user engagement, retention, and also the satisfaction of suppliers (e.g. artists in the music recommendation platforms) would be influenced.



(a) MovieLens



(b) Last.fm

Figure 5: Suppliers' centric view of popularity bias

6 CONCLUSION

Recommender systems are multi-stakeholder environments; in addition to the users, some other stakeholders such as the supplier of the items also benefit from the recommendation of their items and gaining a larger audience. The algorithmic popularity bias can negatively impact both users and suppliers on a recommender system platform. In this paper, we demonstrated the severity of the popularity bias impact on different sides of a recommender system using several recommendation algorithms on two datasets. We also proposed metrics to quantify the exposure bias from the perspective of both the users and suppliers. Our experiments showed that when the recommendations are calibrated for the users in terms of