POPULARITY BIAS IN SEQUENTIAL RECOMMENDATION

A Research Paper submitted to the Department of Computer Science
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By

Shivaen Ramshetty

April 25, 2021

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISOR
Hongning Wang, Department of Computer Science

Popularity Bias in Sequential Recommendation

ABSTRACT

This paper describes the existence of popularity bias within sequential recommendation models to propose methods to such bias. Popularity bias is the phenomena where items that appear more within the data during training or testing are more likely to be recommended without necessarily being the best recommendation. In other words, the work described in the paper defines what fairness is in regards to popularity bias and how a model can be improved to maintain this fairness. In order to do so, we first defined fairness in terms of popularity over time and created a metric that can evaluate such. Additionally, this metric can be used alongside previous baseline metrics to measure bias in sequential models. Then, using the metrics we measured bias in baseline sequential models we found that our metric was capable of illuminating bias effectively and could be applied to debiasing approaches.

ACM Reference Format:
. . Popularity Bias in Sequential Recommendation. In *Capstone Portfolio.* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

Bias and fairness can be defined in many ways, but this paper will focus on a single definition for each. Doing so, will simplify the goal and objectives of the work as well as aid in avoiding any unnecessary confusion.

1.1 Sequential Recommendation

First, it is necessary to understand the basis of sequential recommendation and the reasoning for employing such a method. Sequential recommendation is a form of recommendation system that makes use of sequences of user action to make a recommendation/prediction. Therefore, common models for this form of system are RNN's as well as Markov chains due to their ability to learn these sequential patterns [6]. However, the reason to pay attention to the sequence of actions is the belief that users perform and interact with the data in an ordered fashion. In general, recommendation systems are often content-based and a user will act on that content through views, clicks, purchases, ratings, etc. [7]. With such behavior, sequential data would allow to learn and recognize patterns within the set of data through the order of actions taken. For example, a product bought may often lead to another complementary purchase, but this may not hold true the other way around; patterns where one action leads to another are those which sequential models hope to find.

1.2 Bias and Fairness

Bias in recommendation systems arises from a variety of factors and can be used to describe many types of recommendation behavior. For the purpose of this paper, we will focus on the causes and effects of popularity bias. Popularity in recommendation systems refers to how common an item within the data is interacted with. Hence, popularity bias is where popular items are more often recommended compared to the rest of items in the item set [9, 10]. Such a possibility suggests that when an item that is more popular is recommended, other possibly better fitting items may not be chosen. That is, an item being popular is not directly correlated to that item being the best recommendation to a particular user. Therefore, it is important for us to be able to distinguish popularity bias in models and address the problem if it exists. Then, fairness is another side to consider when evaluating our predictions or next recommendation. Fairness can be defined in respect to certain goals, such as meeting user expectations known as calibration fairness [3]. Another type of fairness is the equality of opportunity, where the goal is to provide an equal chance for each item to be recommended or learned upon. Fairness in terms of opportunity is closely tied to our goal to restrict popularity bias, since the aim is to allow items with less popularity to have a relatively equal chance to be recommended to a user. However, the notion that all popular items should be punished is another extreme, since the goal is to meet the needs of as many users as possible. We do not solely want to address the needs of one group of users while harming another, but in terms of fairness some groups will be less advantaged when prioritizing different aspects of a systems [3]. Lastly, fairness and bias are adapted to different scenarios as technologies change and new innovations come about; therefore, it is important to understand what the application we are in pursuit of and if a definition exists that accurately and justly fits the use case of a project.

1.3 Why study bias in sequential systems?

Throughout the history of machine learning, no matter it be simple algorithms or the extents of deep learning, bias has been a known property of the science. Each system's data will influence the model in one way or another, and changing or diversifying the data will only alter the extent or type of bias. So, what makes sequential bias different? In previous studies of bias, it was common to experiment and observe the results after training, since the model was concrete. When a model is concrete, one knows the bias in the model and how it is going to effect the outcome every time. Furthermore, this bias is not changing, which can be seen as both a positive and negative side-effect. For instance, a model cannot learn worse biases after training and being tested on a very small subset of data, but it also cannot reduce the effect of its bias either. Now, with sequential models, bias does not stop changing at the end of training, as each point in the testing data also affects bias. What this means is that a sequential model at the end of testing may behave completely different to the same model after n steps of testing, due to the testing introducing different types or limiting the impact of bias.

1.4 Temporal Popularity Fairness

With the vast and nuanced definitions of bias and fairness we have focused on a subset within this paper. The aforementioned popularity bias will be tied into a time dependent evaluation, which is the

Technical Report UVa CS, Spring, 2021, Charlottesville, VA
.

fusion of both popularity and temporal biases [10]. However, in this case the temporal aspect will also serve to illustrate and quantify the ever-changing sequential model. This time element is unique to sequential models similarly to the previous differences described above, but time may not be the variable in other applications. To summarize, our fairness definition seeks to assess popularity bias at every time interval, in order to gauge the variance in popularity fluctuation among items and their associated ranks.

2 RELATED WORK

2.1 Models

Recommendation systems have grown in popularity and applications stemming from the growth of content platforms as well as the online ecosystems. Recently, deep learning has lead to great innovation of recommendation systems, since models can know learn more about the users [12]. Additionally, deep learning has opened doors for new designs for model architectures as well as the system itself, which has enabled recommendation system to further develop their accuracy and breadth of application. Finally, deep learning methods are varied and are suitable for certain use cases, due to the manner sequential nature of our work, RNN's are the technique applied.

The RNN used within this paper was introduced and implemented in [8], known as GRU4Rec, it employs layers of gated recurrent units with feed forward layers before outputting. GRU4Rec outperformed baselines such as BPR-MF and POP, illustrating a continuous advancement of RNN utilization in recommendation systems. Interestingly, [8] utilize the RNN for "session-based" recommendation, which is a sub-type of sequential recommendation as a whole [6]. Sequential recommendation also over arches session-aware systems, where the session can be attributed to a user unlike session-based systems. In our case, using the Taobao and Xing data sets, we are able to study session-aware recommendation; however, since our study does not give importance to the user, we can omit user information and test as though the RNN is being applied to a session-based system.

2.2 Popularity Bias

Popularity bias has been studies in a variety of settings, such as in [2], where they focused on the long-tail items in their data. Changing or adapting the definition of popularity bias is quite common, for example in [11] popularity bias is attributed directly to ranking position of items and they use a global popularity outlook. The paper uses averages of item recommendation frequency collected from groups of item which were split by their training popularity. Hence, it is clear that defining popularity seems to be a key aspect of previous papers, especially to observe a specific phenomena. One common distinction made in popularity bias papers, is the difference between whether or not global or local popularity values are accounted for. In other words, is an item's popularity continually increase over the entirety of the data set or does it get counted up and reset at certain intervals. The previously mentioned global popularity paper counts across the entire training set with no intervals. However, [11] sample testing to be uniform distribution of items which is why they only count popularity values for training. Therefore, the choice of popularity definition affects the implementation of the model and allows for studies to have different perspectives on the same problem.

2.2.1 *Problems.*

So, what makes popularity bias bad? [1, 4] state that some problems are:

1. **Decreases Personalization and/or Item Diversity**: This occurs when items are "hidden" or not recommended to users since higher popularity ones take priority. Then, users are selecting from a small subset of items that may leave less popular items unseen.
2. **Popular does not suggest Quality**: An item that is popular may not be the highest quality recommendation. Meaning, a better fit for a user may exist that has a lower popularity but will not be recommended. Such a scenario resembles the previously mentioned failure of biased systems in meeting the needs of different groups.
3. **Popularity may lead to a positive feedback loop**: As popular items are recommended, the user interacts with these same items thereby reinforcing the popularity. Over time, less popular items are drowned out and the first problem grows, where fewer and fewer items are important to the model.

2.3 Fairness Metrics

Various metrics have been implemented to measure the plethora of fairness definitions that exists. Zhu et al. measure the correlation between popularity and ranking using two metrics called *PRU* and *PRI* [13]. *PRU* measures an expectation of popularity and ranking correlation for a user while *PRI* measures the correlation between an item's popularity and its average rank. The goal of this paper was to measure popularity-opportunity bias built off the ideas of popularity bias and equal opportunity. They define equal opportunity as the case when groups have equal true positive rates, meaning that each group should have an equal proportion of correctly recommended items, and in this case those groups would be defined by the popularity of each item. Other works such as Mehrabi et al. have thoroughly identified types of fairness and given definitions to them, with equal opportunity again being described in terms of true positive rates between groups [10]. However, a distinction in this paper is the use of protected and unprotected groups to distinguish the difference between two sets of data. This idea has been extrapolated to multiple groups and therefore more than 1 of each can exist. Another type of fairness mentioned is equality of odds, where both protected and unprotected groups should have the same true positive and false positive rates [10]. Some other metrics include *ARP* and *APLT* from [2], where the former calculates the average popularity of recommended items while the latter evaluates the percentage of long-tail items in the recommended lists. Both are useful in understanding either the overall popularity unfairness of the model or the model's usage of long-tail items. To clarify, long-tail items refers to items who fall behind the most popular groups of items but are also not the least popular. They lie somewhere in between and [2] suggest that they are critical for business to recommend in order to maintain the diversity of items as mentioned above.

3 METHOD

We utilize the GRU4Rec to first confirm popularity bias in both the Taobao and Xing datasets. To do so, we get global popularities of each item and group them using kmeans. Additionally, to determine if the bias existed we used common performance metrics: recall@k and mrr@k.

- **Recall@k**: Calculates the proportion of relevant items from the k item predictions for each sequence. Closer to 1 is better and values range from 0 to 1.
- **MRR@k**: Calculates the average reciprocal rank of all predictions made, where rank is within k k item predictions. Closer to 1 is better since it suggests that the target/ground truth item was recommended near the top of the k predictions more often. Values for mrr range from 0 to 1.

If popularity bias exists, we expect to see that as popularity buckets with larger popularity items to perform better in terms of recall and mrr. However, this has already been performed in previous studies, so we shifted to observing the same through the use of our sequential model. What this means is that we also have the dimension of time. Time helps establish the order of sequences, so we compute the same popularity grouping as before but across multiple time periods. Our time periods were defined on the Taobao dataset and were chosen to equally distribute the data into intervals. However, it is quite simple to convert this into a process where a time interval is rather x number of sequences.

Taking a different angle on this popularity bias, we also visualize the amount of recommendations for each item and the popularity of that item. Doing so shows any relationship between recommendation probability and popularity, where a positive relationship intuitively suggest popularity bias. Finally, we can take a look at the probability of certain item recommendations given their popularities to determine bias and fairness. To do this, we assume that a not fully debiased but relatively fair model would have a relationship:

Probability of item predicted in test = Probability of item in training

Such a proportionality means that items popularity in training should be reflected in testing as popularity bias suggests, with higher probability in test when more common in training. However, if we see that that item become even more probable in testing then the bias has actually become worse than this linear relationship.

3.1 Temporal Discounting

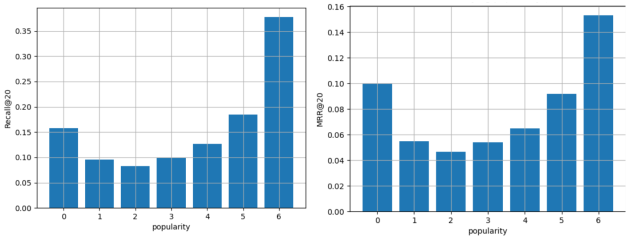
In order to evaluate fairness we created a new metric that is particular to sequential models. We call this metric temporal discounting and the idea is to evaluate a factor that is representative of popularity variance as well as false positive rates across time periods. First, we compute kmeans on popularity groups at time intervals determined by a set number of sequences rather than preset times. During testing, we compute false positive rates for each item and attribute this to each popularity group the item belonged in all past times, these rates we call *Pr*. Thereby, we show the impact of testing results from the popularity distributions earlier in the model. Finally, we multiply a weight W to variance of the rates of each popularity group for every time and sum the products. Doing so, allows the implementer discretion in giving importance to certain time intervals over others, therefore the discounting aspect of the name. For example, by weighting the early periods minutely, their impact is discounted or considered less impactful to the current rates or predictions. Lastly, the factor can be used alone or in unison with previous metrics such as *PRI*. Given number of times n and number of popularity groups g, we calculate the temporal discounting factor (TDF) as follows:

$$TDF = \sum_{t=1}^{n} w_t var(Pr_{1t}, ..., Pr_{gt}) \tag{1}$$

Pr_{gt} represents the rate of popularity group g at time t. We use this metric alongside the aforementioned *PRI* as a baseline to evaluate our debiasing methods.

4 EXPERIMENT

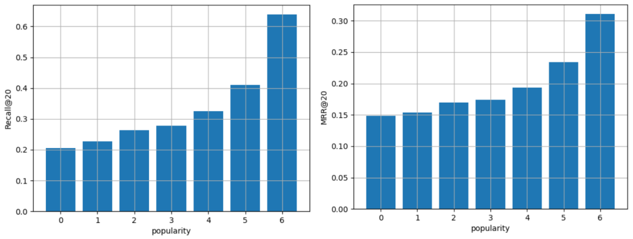
4.1 Confirming Popularity Bias



(a) Recall@20

(b) MRR@20

Figure 1: Bar plots of RNN performance on Taobao test data.



(a) Recall@20

(b) MRR@20

Figure 2: Bar plots of RNN performance on Xing test data.

We first graph the relationship between popularity groups and their respective performance, using both recall and mrr, in Figures 1 and 2. Popularity groups were created using kmeans clusters, where group 0 represents items with the least popularity and group 6 represents the most popular items. Then, we observe that a positive relationship exists between popularity and performance, since an increase in performance is clearly depicted alongside increasing popularity.

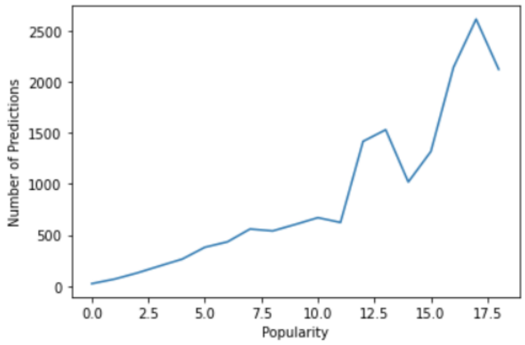


Figure 3: Comparison of the number of times RNN predicts a certain item and that item's popularity.

Following this, we graphed the average number of recommendations/predictions for items and their associated popularity in Fig 3. The x-axis represents the popularity of an item divided by 10, due to number of recommendations being averages of items within 10 popularity apart. Therefore, an x-axis value of 12, translates to items that were in ranges of popularities 120 to 129. Both these tests clearly demonstrate that popularity correlates to better model performance, in other words we verify popularity bias.

4.2 Popularity Fairness

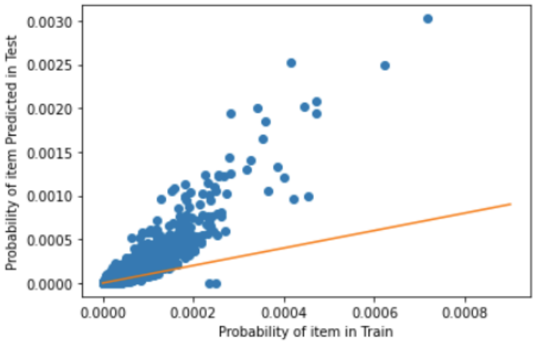


Figure 4: Comparison of the number of times RNN predicts a certain item and that item's popularity.

As described, we made experimented on popularity fairness by first defining a fairness/reasonable curve, this curve is seen in orange in Fig 4. Afterwards, we observed the pattern items follows in terms of prediction popularity in testing, if they follow the the linear trend of the orange line then we can say that testing in sequential models mimics that of training. However, it is clear that testing shows higher rates of popular training items, suggesting that popular items continued to not only influence the model during testing but also had a larger impact.

4.3 Metrics

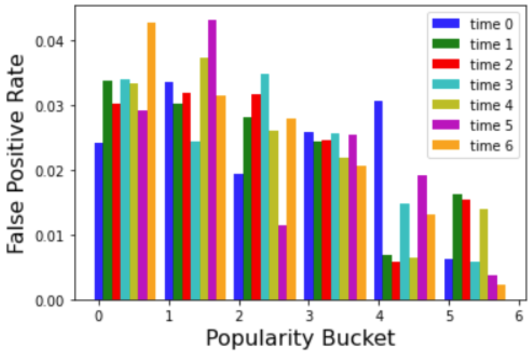


Figure 5: Comparison of false positive rates for different popularity buckets over time. Since popularities of items changes over time, we expect and see varying rates in different time intervals.

Lastly, we utilized **PRI** and our temporal discounting to evaluate numerically the bias or unfairness of the model. In Fig 5, we see the distribution of rates from out temporal discounting method to illustrate the variance over time. We calculate the rates based off of testing data rates that associated to the item in question's popularity bucket, in particular this process could be iterative through intervals of testing or in online learning. Part of temporal discounting is defining weights, we used 5 times and 6 popularity buckets, so we defined two different functions to create weights for each of the 5 times. One equation is exponential, thereby discounting early or furthest away from current time heavily, while the other is a logistic model.
Exponential:

$$w_t = e^{(1/1.5t)} - 1 \tag{2}$$

Logistic:

$$w_t = xlog(t) + 0.99, \tag{3}$$

In equation 3, the x represents a chosen rate, which can determine how fast or slow the weight values decay. Using exponential on Taobao testing gave us a discount factor of 0.0003, while logistic resulted in 0.0006.

5 CONCLUSION

In this paper, we were able to establish and introduce some key ideas to a novel area of recommendation systems. We began by showing that popularity bias not only exists in sequential recommendation, but testing has the potential to exacerbate model bias. Furthermore, we illustrated ways in which to find such bias in future applications that involve sequential models.
Additionally, we define what fairness is in the context of sequential models that have some temporal element, whether that be the number of predictions made or time stamps. By developing temporal discounting, we attempt to connect popularity of items through time to account for the variance in the history of a model. By giving weights to each time we then allow for importance to be placed where desired. Generally, a component to improve upon for many of the fairness metrics is computation time, since in our case,

the growing number of times in testing as well as large itemsets meant very long runtimes for temporal discounting and *PRI*. Such a problem may be addressed by completely changing the metric or adapting pieces of the process as needed.

5.0.1 Future Work. Though our work adapts and creates a new way to assess sequential models, it did not extend the metrics to debiasing methods. In doing so, the metrics could be tested and further developed to depict the way our temporal discounting compares to previous metrics. For example, does a certain debiasing strategy change *PRI* far greater than our temporal discounting, or do both move proportionally. If, they move together through various debiasing processes, then it would be fair to conclude that the two are very similar, however since the fundamentals are quite different, we would expect some differences.Therefore, debiasing could further the current work into the innovation of new models and methods that improve the fairness of sequential models.
Finally, we chose popularity as the focus element of our research, from bias to new metric; however, popularity is not the only measurable and useful descriptor of data items. Studies in the future could extrapolate work to sequential models that are time on elements such as temporal patterns in data as mentioned in [5].

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning to Rank Recommendation. *SOCRS*.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing popularity bias in recommender systems with personalized re-ranking. *AAAI Florida Artificial Intelligence Research Society* (May 2019).

[3] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *RecSys* (2019).

[4] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: a survey and future directions. *Transactions on Knowledge and Data Engineering* 1, 1 (Jan. 2020). https://doi.org/10.1145/3426723

[5] Nan Du, Yichen Wang, Niao He, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. *Advances in Neural Information Processing Systems* 28.

[6] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems* 1, 1 (Jan. 2020). https://doi.org/10.1145/3426723

[7] Vijay N. Gadepally, Braden J. Hancock, Kara B. Greenfield, Joseph P. Campbell, William M. Campbell, and Albert I. Reuther. 2016. Recommender systems for the department of defense and intelligence community. *Lincoln Labratory Journal* 22, 1 (Jan. 2016).

[8] Balazs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. *ICLR* (March 2016).

[9] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback loop and bias amplification in recommender systems. *CIKM'20: International Conference on Information and Knowledge Management,* (Oct. 2020).

[10] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. (Aug. 2019).

[11] Tianxin Wei, Fuli Feng, Jiawei Chen, Chufeng Shi, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2020. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender systems. (Oct. 2020).

[12] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2018. Deep learning based recommender system: a survey and new perspectives. *ACM Comput. Surv* 1, 1 (July 2018).

[13] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity bias in collaborative filtering. *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining.* https://doi.org/10.1145/3437963.3441820