Alleviating the unfairness of recommendation by eliminating the conformity bias

Zhe Ma¹ and Qiang Dong*¹

¹School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China
*Corresponding author’s e-mail: dongq@uestc.edu.cn
*Corresponding author’s ORCID: https://orcid.org/0000-0003-1986-8961

Abstract. Recommendation systems play a critical role in the Internet era. In the past decade, many methods have been proposed to improve the accuracy of recommendation. However, simply increasing the accuracy is usually accompanied by the popularity bias in the recommendation results. In addition, from the perspective of item providers the popularity bias will bring unfairness in exposure of different items to users. In this paper, we propose a method to alleviate problems of unfairness and popularity bias. The main idea is based on the hypothesis that some interactions of users with popular items are not due to their preferences, but attribute to the conformity propensity, which aggravates the popularity bias problem. Thus we consider to eliminate these so called “unreliable interactions” to alleviate the unfairness. We propose a two-step preferential diffusion process between conformity users and popular items in the random walk recommendation algorithm to reduce the effect of unreliable interaction and alleviate the popularity bias and unfairness under the condition of ensuring accuracy. The experiment results on the two benchmark datasets, Movie-Lens and Netflix, show that our algorithm can improve the fairness of recommendation results and alleviate the popularity bias while achieving slightly increased recommendation accuracy.

1. Introduction

Recommendation systems have been widely used in many industrial fields to alleviate the problem of information overload caused by the development of the Internet. In the early stages of the development of recommendation systems, in order to improve the experience of the user, the accuracy of the recommendation algorithm has been paid more attention. Researchers have used many methods to improve the accuracy[3,4]. But now researchers have found that simply improving the accuracy of the recommendation list of users cannot improve the user experience. By contrary, blindly improving the accuracy is usually accompanied by the popularity bias that too many popular items appear in the recommendation list and these low-informative popular items will reduce the user experience. Besides, from the perspective of product providers the problem of popularity bias will lead to unfairness for the item providers of unpopular items because it is difficult for the unpopular items to have a chance to be exposed[2]. Popular items account for the vast majority of the recommendation list, while those newly joined unpopular items rarely have the opportunity to be recommended, so it is difficult for these merchants to make a profit, which will have a negative impact on the platform in the long time. Therefore, the work of correcting the popularity bias and unfairness in recommendation has gradually become a hot topic. However, as we all know, the accuracy is trade-off fairness[3]. How to reduce the popularity bias while ensuring the accuracy in a high level has become difficulty in the
recommendation system. Various methods have been proposed to solve the problem of popularity bias. Krishnan et al[11] found that user ratings follow different distributions when users rate items before or after being exposed to the public opinions. Conformity bias happens in the user interaction, making the rating values do not always represent user’s true preference.

Inspired by conformity factors, this article incorporates the conformity factors into the random walk based recommendation algorithm. For some interactions of popular items in the dataset, it is believed that not all interactions are generated by the user personal preferences, and some are caused by conformity factors. For example, when a user knows a popular product, although the user has little preference on the product, the curiosity and conformity caused by the popularity of the item make the user purchase it. The interaction caused by conformity factors and the interaction caused by the personal preferences of user should not play the same role in the calculation process of the recommendation algorithm. Therefore, in this article we propose a solution from the perspective of unreliable interaction. The method is validated on two benchmark data sets and find that the algorithm proposed in this paper can greatly improve the fairness metrics and diversity metrics under the premise of ensuring the high recommendation accuracy.

2. Preliminary knowledge

2.1. Notations
In this paper, all of our models are based on random walk algorithm which are the kind of method abstracting the recommendation process into resource propagation in a bipartite graph network composed of users and items. The bipartite graph network can be described as $G(U, I, R)$, where $U = \{u_1, u_2, u_3, ..., u_m\}$ denotes the set of user in recommendation, $u_m$ denotes use $m$; $I = \{i_1, i_2, i_3, ..., i_n\}$ and $R = \{r_1, r_2, r_3, ..., r_p\}$ denote the sets of items and links in recommendation respectively, $i_n$ denotes item $n$ and $r_p$ denotes link $p$. In the graph, if the starting node of the link $p$ is user $w$ and the ending node is item $v$, it means that the user $w$ has collected the item $v$.

The bipartite graph also can be described as adjacency matrix $A_{m \times n}$, where the element $a_{xy}$ equals 1 if user $u_x$ has collected item $i_y$, and 0 otherwise. $d_{uk}$ denote the numbers of items collected by user $u_k$, as known as the degree of $u_k$. $d_{it}$ denote the numbers of users who have collected item $i_t$, as known as the degree of $i_t$.

2.2. Related work
For the past few years, many methods have been proposed to solve problems of popularity bias and unfairness. For example, Zhou et al[6,8] introduce P3 algorithm[1] which will be introduced in detail in the next section and a series of methods inspired by common physical phenomena to recommend systems, other researchers have proposed many improved algorithms based on them[7-8]. A representative work is RP3 algorithm[13] which uses the degree of items to reorder the scores generated by P3 algorithm. Because of the attenuation of popular items is greater than unpopular items, the score of unpopular products are relatively improved.

Another related algorithm is the PD algorithm [10]. The difference between PD algorithm and P3 algorithm is that the PD algorithm did not reorder the recommendation results, it changed the way of diffusion, and carried out a biased diffusion for items with different popularity. The diffusion ratio is proportional to the inverse of the popularity of the items. Although the methods of the two algorithms are of different forms, the essential ideas are the same. Both of them attenuate the score according to item popularity.

Another method, the RA algorithm [4] is a re-rank method. Firstly, prediction score for the item of the user is obtained according to the random walk algorithm. And then we rank the result and get the rank number of the item under the user and the rank number of the user under the item. Finally, we combine these two rank numbers with weights, and then decide whether the product is recommended to the user based on the combined rank number. This method takes into account the preferences of
items for the same user and the rank of users for the same item, so it can be ensured that the items recommended to users are items that users like. At the same time, rank numbers are used instead of scores. Since the scores of unpopular products are generally not high compared with popular items, using rank can improve the probability of unpopular products being recommended.

There are also a class of methods that transform the recommendation problem into an allocation problem \[5, 12\]. Firstly, we get the user preference for all items by scoring algorithm and determine the length of the recommendation list, then iteratively assign one item to each user at a time. It is determined whether to allocate the item to the user based on the preference score of the commodity and the information of the allocated items in the current recommendation list. For example, this algorithm divides all items into long-tail items with low popularity and head items with high popularity. When items are assigned to users, not only the user preference for the item and category, the proportion of the category in the current recommendation list must also be taken into account[5]. Another algorithm introduces the envy free theory in game theory to restrict the recommendation of popular products[12].

2.3. Baseline algorithm

2.3.1. P3 algorithm. The P3 algorithm, also referred as NBI in other articles is the original random walk algorithm. The procedure of P3 can be divided three steps in user-item bipartite graph network: For a target user, firstly we initialize the unit resource to the items that user has collected. Then spreading resources evenly to user nodes from item nodes in the second step. Finally, as in the second step, the resources are propagated from the user nodes to the item nodes. Ranking the uncollected items in decreasing order according to the distribution of resources on item nodes generated by the above three steps, the top-K items constitute the final recommendation list for the user. We merge the second and third steps then it can be written as a resource transfer matrix \( W \) where the element of \( W \):

\[
W_{ij} = \frac{1}{d_{ij}} \sum_{l=1}^{m} a_{il} d_{il} \alpha_{ij} \tag{1}
\]

Where the \( a_{il} \) equal 1 if user \( l \) collected the item \( \alpha \) meaning that item \( \alpha \) get one unit resource from user \( l \), otherwise 0. So we can get the final distribution of resources according to the following formula:

\[
S = A^*W \tag{2}
\]

Where \( A \) is the adjacency matrix and the element \( \alpha_{xy} \) equals 1 if user \( u_x \) has collected item \( i_y \), and 0 otherwise.

2.3.2. RP3 algorithm. RP3 algorithm is an improved algorithm of P3, which introduce the information of degree of item to revise the score generated by the P3 algorithm to different extent. The formula is as follow:

\[
S = A^*W^*D^{-\lambda} \tag{3}
\]

Where \( D_{ii} \) is the diagonal matrix consisted of the reciprocals of degrees of each item in data set. Each score will be weakened by the degree of the items. Obviously, the greater is the degree of the item, the greater the score of the item was weakened. This means that the score of this popular item has been more weakened. The range of the parameter \( \lambda \) in the formula is from 0 to 1. When \( \lambda \) is set to 0, the RP3 algorithm reduces to the P3 algorithm. With the increasing of parameters, the degree of reduction of popular items has also increased compared with long-tail items.

2.3.3. PD algorithm. The PD is the abbreviation for preferential diffusion. Compared with P3, it replaced even diffusion with preferential diffusion in the third step. The amount of resource that an
item $\alpha$ received is proportional to $d_{i\alpha}^{-\varepsilon}$, where $\varepsilon$ is a free parameter. In this case, the resource transfer matrix reads:

$$w_{\alpha\beta} = \frac{1}{d_{\beta}^{-\varepsilon}} \sum_{i=1}^{n} \frac{d_{i\alpha}d_{ij}}{M}$$

(4)

Where $M$ indicates the sum value of $d_{i\alpha}^{-\varepsilon}$ which refers to the reciprocal of degree of items collected by user $u_t$. Where $\varepsilon$ is a free parameter.

3. Method

In this section, we will introduce our model in detail. Firstly, we introduce the unreliable interaction which are the critical of concept in our proposed method. Then, we will introduce how can we use those unreliable interaction to alleviate the popularity bias and unfairness problem.

We can abstract the recommendation system into three parts: Data, Model, User interaction. Recommendation model has been trained by the data organized by user interaction, then the trained recommendation model can serve the user by generating the item list. The above process is a recommendation loop. Bias in each part of the recommendation loop will lead to the bias in recommendation results.

Our work focus on the phase from user interaction to data. In this phase, we collect and organize the user interactions into a feedback matrix. In terms of the form of data, feedback matrix can be divided into implicit matrix and explicit matrix. The implicit matrix, where each entry is binarized to indicate whether the user has interacted with the item; and explicit feedback, where each entry is real value that directly reflects user preference on the items.

3.1. Define of concept

Conformity bias happens as users tend to rate similarly to the others, even if doing so goes against their own judgment, making the rating values do not always signify user true preference[9]. Krishnan et al. [11] found that user ratings follow different distributions when users rate items before or after being exposed to the public opinions. Hence, the observed ratings are skewed and might not reflect user real preference on items.

From the above, user interaction can be divided two types: implicit feedback and explicit feedback. Obviously, conformity bias will also occur in implicit feedback. We made the hypothesis that popular items will get more click or purchase by users due to their own popularity. So those interactions among popular item cannot reflect user preference, the interaction with popular items may be due to conformity propensity of users. Therefore, based on this hypothesis, we define those user interactions that cannot reflect the real preferences of themselves as unreliable interaction. These unreliable interactions should play a smaller role than reliable interaction.

Now the most important thing is how can we find those reliable interaction. Here we give our solution for this problem. The intuitive idea is that if the historical interaction of user has a high proportion of popular products or the average degree of items in historical interaction of user is large, it is considered that the user has a high probability of being a conformity user. In addition, if the popularity of the item among this interaction is very high, this interaction is likely to be an interaction caused by other factors, such as conformity. Therefore, deciding whether it is an unreliable interaction needs to be considered from two aspects. One is the user degree of conformity, which is reflected in the data as the average value of historical interactive items of user, and the other is the popularity of the item among this interaction.

3.2. Our model

Now we have already understood the above definition, so we consider how to weaken the role of unreliable interaction in the recommendation algorithm. Inspired by the PD algorithm, our model introduces the preferential diffusion into the second diffusion step on the basis of the preferential diffusion in the third step of the PD algorithm. In PD algorithm, the proportion of resource spreading
to popular items nodes is weakened. In our method, the conformity is considered to the preferential diffusion in the second step of the random walk. The preferential diffusion to the conformity user similarity to the PD algorithm is to reduce the proportion of resource spreading to the user node with high average degree of the historical interaction. By analogy, the larger the average degree of item in historical interaction of the user, the greater the degree of conformity. Therefore, the resource transfer matrix is as follows:

$$w_{\alpha\beta} = \frac{1}{N\sum_{i=1}^{m} a_{\alpha i} d_{i\beta}} M$$

$$M = \sum_{i=1}^{m} a_{\alpha i} d_{i\beta}$$

$$N = \sum_{i=1}^{m} a_{\alpha i} c_{i\beta}$$

$$c_{i\beta} = \frac{1}{k_{i\beta}} \sum_{e=1}^{n} a_{\alpha e} d_{e\beta}$$

Where $M$ indicates the sum value of $d_{i\beta}$, which is the degree of items collected by user $u_i$. $d_{i\beta}$ indicates the mean value of the degree of items have been collected by user $u_i$. $c_{i\beta}$ refers to the conformity degree of user. The larger its value, the greater the conformity degree of this user, which is proportional to the average degree of the historical click items of user. $N$ indicates the sum value of $c_{i\beta}$. Where $a_{\alpha e}$ equals 1 if user $u_e$ has collected item $i_e$, and 0 otherwise. When the $c_{i\beta}$ equal 0, it gives the PD algorithm.

4. Experiment

In this section, we first show our experiment procedure, including the datasets, the evaluation metric and the parameter selection. Then we will present the experiment results at the end.

4.1. Dataset

Movie-lens and Netflix are two benchmark dataset that are widely used in recommendation. They are all movie rating dataset. For movie-lens dataset, it has many versions with different scale. In our experiment Movie-lens 1m dataset was picked as our dataset out of computing resources. The Movie-lens 1m dataset contains 836477 links which are consists of 6038 users and 3628 movies. Each link has four components (user, item, score, timestamp), means that the user rated the movie at the time point indicated by the timestamp. The rating ranges five level from 1 to 5, reflecting the user preference for the movie. The larger the value, the greater the preference of user for the movie. In our experiment, we retain the links with rating equal or greater than 3 as positive samples, then remove the links with rating less than 3. After data filtering, we need to ensure there are no isolated nodes in the filtered data set because isolated nodes cannot be recommended in algorithm. The Netflix dataset is a random sampling of the whole links of user activities in Netflix.com. It consists of 10000 users, 6000 movies and 824802 links. Similarity to the Movie-lens dataset, only the links with ratings no less than 3 are considered.

4.2. Evaluation metric

Firstly, the most important metrics to evaluate the performance of the recommendation algorithm is the accuracy metrics which indicates how much the users like items in the recommendation list. In our experiment we used precision as accuracy metrics to measure the performance of algorithm. The algorithm will recommend a fixed number of items for each user known as recommendation list. In the formula of precision the length of the recommendation list is as the denominator, and the number of hit items (items that were recommended to user also appear in the user test set) is as the numerator. So
the precision is the fraction of hit recommended items to the length of recommendation lists. Then averaged over the precision values of all users, we obtain the precision of the recommendation system. The formula of precision is as follows:

\[
\text{Precision} = \frac{1}{m} \sum_{k=1}^{m} \frac{h_{uk}}{l}
\]  

(9)

Where the \(h_{uk}\) refers to the number of hit items in recommendation list of user \(k\), and \(l\) refers to the mixed length of recommendation.

In addition, the coverage rate was used to measure the performance of recommendation result in fairness. Coverage rate indicates the proportion of items in the recommendation result to the entire item pool of the recommendation system. The larger the value, the better the fairness of the recommendation. The formula of precision is as follows:

\[
\text{Coverage} = \frac{o}{n}
\]

(10)

Where \(o\) refers to the number of items in recommendation list of all user.

However, coverage rate can only reflect the proportion of items, the times of the items appear in the recommendation list is not be taken into account, so sometimes the robust of coverage rate cannot perform well. In order to evaluate the fairness of the recommendation result better, we use the Gini coefficient to measure the balance of the numbers of times of different items in recommendation list. It not only take into account the proportion of items appearing in recommendation result, but also the frequency of items which are in recommendation list. The formula of Gini coefficient is as follows:

\[
\text{Gini} = 1 - \frac{1}{n-1} \sum_{k=1}^{n} (2k - n - 1)p(i_k)
\]

(11)

Where \(p(i_k)\) refer to the \(k\)-th least of the proportion of item in the recommendation list to recommendation result of all items.

Besides the above two accuracy and fairness measures, in this paper, the diversity of recommendation list is another aspect we should consider. The difference of recommendation list of each other represents the level of personalization of the recommendation algorithm. Hamming distance is popular method in evaluating the difference of the recommendation list. The formula of Hamming distance is as follows:

\[
\text{Hamming} = 1 - \frac{1}{m \times n} \sum_{x=1}^{m} \sum_{y=1}^{m} \frac{\text{diff}(L_{ux}, L_{uy})}{l}
\]

(12)

Where the \(\text{diff}(L_{ux}, L_{uy})\) refers to the number of items which co-appeared in the recommendation list of \(u_x\) and \(u_y\).

4.3. Results

In our experiment, we split the original dataset into train set containing of 80% of data and test set containing 20% of data and the length of recommendation list is set to 20 for all the algorithms. In terms of parameter selection, for RP3 algorithm and PD algorithm, we choose the parameter with the highest accuracy as the optimal parameter because of accuracy metrics is most important, so the optimal parameter \(\varepsilon\) of the PD algorithm is -0.85 which is consistent with the parameters in the original paper. The optimal parameter \(\lambda\) of RP3 algorithm is selected as 0.7. For the determination of the parameters of the algorithm TS-PD proposed in this article, it is necessary to ensure that the accuracy metrics of the TS-PD algorithm is not worse than the above baseline algorithm, and on this basis, chose the parameters which can improve the fairness and diversity metrics as much as possible.
Table 1. Experiment results on Movie-lens.

| Algorithm | Gini  | Hamming distance | Coverage rate | Precision  |
|-----------|-------|------------------|---------------|------------|
| P3        | 0.01049| 0.51867          | 0.05384       | 0.17084    |
| RP3       | 0.08434| 0.87310          | 0.58784       | 0.24577    |
| PD        | 0.09171| 0.89480          | 0.62531       | 0.24786    |
| TS-PD     | 0.12326| 0.9293           | 0.67583       | 0.24872    |

Table 2. Experiment results on Netflix.

| Algorithm | Gini  | Hamming distance | Coverage rate | Precision  |
|-----------|-------|------------------|---------------|------------|
| P3        | 0.00670| 0.52080          | 0.04044       | 0.13760    |
| RP3       | 0.04246| 0.70263          | 0.67676       | 0.15461    |
| PD        | 0.07099| 0.70672          | 0.79521       | 0.14976    |
| TS-PD     | 0.10092| 0.80863          | 0.84945       | 0.15711    |

Based on this criterion, the optimal parameter of our method $\varepsilon_1$ and $\varepsilon_2$ were selected as -0.85, -0.6 for Movie-lens and -0.85, -0.9 for Netflix.

Based on the above parameter settings, we get the experiment results and summarizes all detail values in Table 1 and Table 2 correspond to the results on Movie-lens and Netflix respectively.

Figure 1. The Precision performance of TS-PD algorithm against the parameter on Movie-lens.

Figure 2. The Precision performance of TS-PD algorithm against the parameter on Netflix.

Figure 3. The Gini Coefficient performance of TS-PD algorithm against the parameter on Movie-lens.

Figure 4. The Gini Coefficient performance of TS-PD algorithm against the parameter on Netflix.
Figure 5. The Hamming Distance performance of TS-PD algorithm against the parameter on Movie-lens

Figure 6. The Hamming Distance performance of TS-PD algorithm against the parameter on Netflix.

Figure 7. The Coverage rate performance of TS-PD algorithm against the parameter on Movie-lens.

Figure 8. The Coverage rate performance of TS-PD algorithm against the parameter on Netflix.

Whether on Movie-lens or Netflix datasets, our proposed method outperforms the other three methods on both accuracy and fairness indicators. On the basis of a slight increase in precision, our method has increased by 34.4% compared with PD algorithm and 46.14% comparing with RP3 algorithm on the Gini Coefficient in Movie-lens dataset. In Netflix dataset our method has increased by 42.16% compared with PD algorithm and 137.68% comparing with RP3 algorithm on the Gini Coefficient. In addition, on hamming distance compared with the 0.8948 of the PD algorithm and 0.8731 of the RP3 algorithm in Movie-lens, the 0.70672 of the PD algorithm and 0.70263 of the RP3 algorithm in Netflix, our proposed algorithm has reached 0.9293 and 0.80863.

4.4. Analysis

In order to evaluate the improvement effect after adding the preferential diffusion to conformity users in second step of PD, we fix the parameter $\varepsilon_1$ and observe the performance of Gini coefficient and precision under different value of parameter $\varepsilon_2$. It can be found in the figure 1-8 that when the absolute value of parameter $\varepsilon_1$ is small, equal 0.7 and 0.75, as the absolute value of parameters $\varepsilon_2$ and the degree of preferential diffusion to conformity user increases, both of accuracy metrics and fairness metrics increase. However, with the increase of the absolute value of $\varepsilon_1$, more than 0.8, when we change the degree of preferential diffusion to conformity user according decreasing $\varepsilon_2$, although the fairness and diversity indicators such as the Gini coefficient and hamming distance increase, the accuracy indicators begin to decline. From the above phenomenon, we can draw the conclusion that preferential diffusion to conformity users will alleviate the problem of popularity bias and fairness in recommendation. Besides, it can improve accuracy to some extent.
5. Conclusion
In this paper, in terms of discovery that user ratings follow different distributions when users rate items before or after being exposed to the public opinions by other researcher, we proposed a hypothesis about conformity bias that there are some interactions with popular items which are due to conformity rather than personal preference factors. We introduce this information as a basis for preferential diffusion into the random walk algorithm. According to experiment result, our method greatly improves fairness metrics on the basis of a slight increase in accuracy metrics. General speaking, our method performs well comparing with other state of the art algorithm. So we can draw the conclusion that, to a certain extent weakening the role of unreliable interaction in the recommendation algorithm can help reduce the recommendation of popular items and alleviate the fairness problem without losing the accuracy of the recommendation.

References
[1] Fleder, D., Hosanagar, K. (2009) Blockbuster culture’s next rise or fall: the impact of recommender systems on sales diversity. Management Science, 55: 697-712.
[2] Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A. (2013) Recommender systems survey. Knowledge-Based Systems, 46: 109-132.
[3] An, Y. H., Dong, Q., Sun, C.J., Nie, D.C., Fu, Y. (2016) Diffusion-like recommendation with enhanced similarity of objects. Physica A: Statistical mechanics and its applications, 461: 708-715.
[4] Dong, Q., Yuan, Q., Shi, Y. B. (2019) Alleviating the recommendation bias via rank aggregation. Physica A: Statal Mechanics and its Applications, 534: Art. No. 122073.
[5] Patro, G. K., Biswas, A., Ganguly, N., et al. (2020) Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In: Proceedings of The Web Conference 2020. Taipei. pp. 1194-1204.
[6] Zhou, T., et al. (2007) Bipartite network projection and personal recommendation. Physical Review E, 76: Art. no. 046115
[7] Zhu, X., Tian, H., Zhang, T. (2018) Symmetrical information filtering via punishing superfluous diffusion. Physica A: Statistical Mechanics and its Applications, 508: 1-9.
[8] Zhou, T., Kuscsik, Z., Liu, J. G., Medo, M., Wakeling, J. R., Zhang, Y. C. (2010) Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences of the United States of America, 107: 4511-4515.
[9] Chen, J., Dong, H., Wang, X., et al. (2020) Bias and debias in recommender system: A survey and future directions unpublished.
[10] Lü, L.Y., and Liu, W.P. (2011) Information filtering via preferential diffusion Physical Review E, 83: Art. No. 066119.
[11] Krishnan, S., Patel, J., Franklin, M. J., and Goldberg, K. (2014) A methodology for learning, analyzing, and mitigating social influence bias in recommender systems. In: RecSys. Silicon Valley. pp.137-144.
[12] Abdollahpouri, H., Robin, B., and Mobasher, Bamshad. (2019) Managing popularity bias in recommender systems with personalized re-ranking. In: The thirty-second international flairs conference. Sarasota.
[13] Christoffel, F., Paudel, B., Newell, C., Bernsteinnaet, A. (2015) Blockbusters and Wallflowers: Accurate, Diverse, and Scalable Recommendations with Random Walks. In: In Proceedings of the 9th ACM Conference on Recommender Systems. Vienna. pp. 163-170.