The Impact of a Popularity Punishing Hyperparameter on ItemKNN Recommendation Performance

Robin Verachtert[1,2](✉), Jeroen Craps[1], Lien Michiels[1,2], and Bart Goethals[1,2,3]

[1] Froomle NV, Antwerp, Belgium
robin.verachtert@froomle.com
[2] University of Antwerp, Antwerp, Belgium
{lien.michiels,bart.goethals}@uantwerpen.be
[3] Monash University, Melbourne, Australia

Abstract. Collaborative filtering techniques have a tendency to amplify popularity biases present in the training data if no countermeasures are taken. The ItemKNN algorithm with conditional probability-inspired similarity function has a hyperparameter α that allows one to counteract this popularity bias. In this work, we perform a deep dive into the effects of this hyperparameter in both online and offline experiments, with regard to both accuracy metrics and equality of exposure. Our experiments show that the hyperparameter can indeed counteract popularity bias in a dataset. We also find that there exists a trade-off between countering popularity bias and the quality of the recommendations: Reducing popularity bias too much results in a decrease in click-through rate, but some counteracting of popularity bias is required for optimal online performance.

Keywords: Recommendation systems · AB test · Nearest neighbour

1 Introduction

Collaborative filtering algorithms are widely used for recommendation systems. To make predictions of what users may like, they rely on past preferences for items expressed by users. These preferences can, for example, be expressed by interacting with an item. Collaborative filtering methods can suffer from a 'rich get richer' effect when they fail to address the popularity bias in the data. For example, when some items are visited more often by users, the recommendation algorithm is also more likely to recommend them. This bias towards already popular items is generally considered undesirable, and many solutions have been proposed to address this bias [e.g. 1,17,24]. Even some of the earlier works on collaborative filtering were mindful of this inherent popularity bias. When Deshpande and Karypis [7] proposed the ItemKNN algorithm, they added a hyperparameter α to their conditional probability-inspired similarity function with

J. Kamps et al. (Eds.): ECIR 2023, LNCS 13981, pp. 646–654, 2023.
https://doi.org/10.1007/978-3-031-28238-6_56

the explicit purpose of discounting popular items that may otherwise dominate recommendations. Recent works have shown that despite advances in the field, ItemKNN and other nearest neighbour-based methods are still competitive, provided they are well-tuned [9,10,16,22]. Because of their inherent scalability, they remain popular methods in production environments.

In this work, we investigate how different values of the hyperparameter α impact performance and equality of exposure, as a measure of popularity bias, in both offline and online experiments with ItemKNN on three news datasets.

We answer the following three research questions:

- **RQ1:** How does the hyperparameter α impact the equality of exposure?
- **RQ2:** How does the hyperparameter α impact accuracy and CTR results?
- **RQ3:** Do the offline and online results agree?

Our work is done in the context of the popular item-to-item recommendation paradigm, recommending similar items in the context of another item, which we will refer to as *context item*. We focus our work on the news domain, as they have a specific interest in combatting popularity bias for ethical reasons, and, of course, because our partners agreed to perform the online tests discussed in this work. All data processed in these experiments was collected in accordance with GDPR: Users consented to receive personalised recommendations, as well as to have their data analysed and to participate in AB testing.

We find that the hyperparameter α can be used to increase the equality of exposure. Secondly, we find that it is necessary to seek a trade-off between equality of exposure and recommendation quality. We leave a thorough investigation into this trade-off for future work. Finally, we note that our offline and online results do not align due to the inherent popularity bias persisted in the offline evaluation [4].

2 Related Work

Popularity bias has been extensively studied in the context of recommender systems [e.g. 1,17,24]. Although the effect of popularity bias on ItemKNN has been studied [2], to the best of our knowledge, the impact of the hyperparameter α on popularity bias has not. In the original work by Deshpande and Karypis [7], the impact of α is evaluated solely in terms of MRR and HitRate, both accuracy measures. Recent work by Pellegrini et al. [19] suggests that not recommending popular items makes recommendations more personalised and can positively impact the recommender system's performance.

ItemKNN remains a popular and competitive baseline, despite recent advances in recommendation algorithms [9,10,16,22].

Due to their scalability, neighbourhood-based methods such as ItemKNN remain a popular choice in production settings [3,8,15,20]. Therefore, a thorough investigation of how the popularity bias can be countered is of great practical relevance.

Offline and online results often do not correlate [4,11,21], although some works have achieved success [12,18]. Popularity bias is an important factor in this failure to correlate and thus we investigate its impact in this work [4].

3 Experimental Setup

In this work, we focus on the item-to-item recommendation problem. The recommendation system needs to recommend users new items while they are currently visiting an item page on the website. The item the user is visiting is the only information the system uses to generate recommendations.

The dataset $\mathcal{D}$ consists of triplets (u, i, t) where $u \in U$ is the user, $i \in I$ is the item, and $t \in \mathbb{N}$ is the timestamp of when user u interacted with item i. Then the recommendation for user u is a function: $\Phi(\mathcal{D}_u^l)$, where $\mathcal{D}_u$ is the list of items that the user has seen and $\mathcal{D}_u^l$ is the last item that the user has seen.

Algorithm. We use the ItemKNN algorithm, with the similarity between items computed using the conditional probability-inspired similarity function, defined as

$$sim(i,j) = \frac{|\{u|i,j \in \mathcal{D}_u\}|}{|\{u|i \in \mathcal{D}_u\}| \cdot |\{u|j \in \mathcal{D}_u\}|^{\alpha}}$$

Here, i is a context item, j is a target item and α is a hyperparameter that punishes popular items in the similarity computation [7].

Specific values for α can be linked to other similarity measures. When $\alpha = 1$ it provides the same recommendations as the lift similarity measure. In the specific case of item-to-item recommendations, $\alpha = 0.5$ leads to the same recommendations as cosine similarity.

Metrics. To evaluate the exposure of articles, we measure both the item-space coverage and the Gini coefficient as suggested in previous works on evaluation [6,13]. Coverage computes the percentage of the available catalogue recommended at least once during an experiment, while the Gini coefficient gives more insight into the recommendation distribution by measuring the inequalities in the number of recommendations each item in the catalogue receives. To evaluate the accuracy of the recommendations, we measured normalised discounted cumulative gain (NDCG) [14], recall [13] and mean reciprocal rank [13]. For brevity, we report only the NDCG results in this paper. Both other accuracy metrics support the same findings. In online trials, we evaluate the quality of the recommendations by click-through rate (CTR).

Datasets. For our experiments, we use three different newspaper websites as our testing platforms, referred to as NP1, NP2 and NP3. The statistics of online traffic and offline exports on these websites can be found in Table 1. Offline datasets are constructed by selecting events from an eight-day window on the website.

Table 1. Statistics of websites used in the online tests.

Website	Users (per day)	Articles read (per day)	Clicks (per day)	$\|U\|$	$\|I\|$	$\|\mathcal{D}\|$	Gini coeff.
NP1	300K	1M	25K	410 843	2 382	4 049 944	0.79
NP2	200K	800K	14K	234 839	2 404	2 852 956	0.77
NP3	1M	4M	160K	1 215 900	5 531	13 842 991	0.88

Offline Experiments. In our offline experiments, we closely mimic the online setup. The first day of our eight-day dataset is used to make sure that we always have a full day of training data when training a model. The second day is used for optimising other hyperparameters than α. The last six days are used for evaluation. Models are trained, following the online setting, on a single day of training data. During optimisation and evaluation, we expect the model to predict a user's last event between 10 AM and 2 PM on each day, using their second to last event in that window as the context item. The measurements from each of the six evaluation days are averaged and reported in this paper.

As our online tests show three items to the user, we also evaluate the offline metrics on the top three recommendations. We ran our experiments for $\alpha \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. For our online tests, we selected $\alpha \in \{0, 0.2, 0.5, 0.7, 1\}$, as they resulted in different exposure distributions. For brevity, we only report results for these values of α.

Online Experiments. Recommendations were displayed in a horizontal list of three items, just after the end of an article. The models for both the control and treatment groups are re-trained every 15 minutes, using a day of training data. This training window was optimised following the procedure defined by Verachtert et al. [22]. In order to evaluate the impact of α in a real and dynamic environment, we have performed a sequence of trials. In each of these trials, a control group of 75% of the users received recommendations using $\alpha = 0.5$. The treatment group (25% of users) received recommendations using a different $\alpha \in \{0, 0.2, 0.7, 1\}$ for each trial period. As it is not possible to compare the CTR between treatment groups, we instead use the lift in CTR for each treatment group compared to the control group during each trial.

4 Experiments

RQ1: How Does the Hyperparameter α Impact the Equality of Exposure? In Table 2 we show that increasing α leads to higher coverage and to more equal exposure between items. Increasing α from 0.7 to 1.0 does lead to only minor improvements in the Gini coefficient and to a reduction of coverage in two datasets.

In Fig. 1 we look beyond the metrics and inspect how the α hyperparameter impacts how often items are recommended on the NP3 website. Items are sorted by popularity, from most popular to least popular along the x-axis. When α is 0, almost all recommendations are from the most popular items. As the value of the hyperparameter increases, more and more different items are recommended, until the distribution shifts when α is 1, and mostly unpopular items are recommended. This insight explains the slight decrease in coverage for some of the datasets, and why the Gini coefficient did not decrease further when increasing α to the max. These distribution plots, also show that none of the α settings provides true equality of exposure, as the middle section of items is always under-recommended, compared to popular or unpopular items depending on the value of α.

Table 2. Coverage and Gini coefficient results for each of the hyperparameter configurations in the online experiments.

α	Coverage@3 (%)					Gini coeff.				
	0.0	0.2	0.5	0.7	1.0	0.0	0.2	0.5	0.7	1.0
NP1	71	87	94	**97**	95	0.91	0.89	0.83	0.79	**0.76**
NP2	57	78	93	**94**	**94**	0.92	0.90	0.83	0.78	**0.76**
NP3	78	94	97	**100**	99	0.91	0.90	0.80	**0.70**	**0.70**

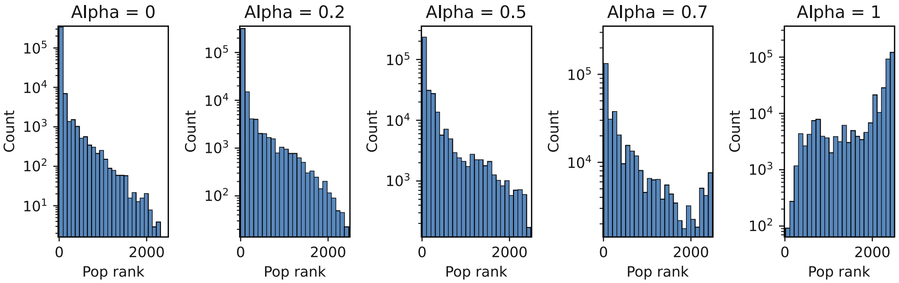

Fig. 1. Number of times items are recommended on the NP3 website experiment, ranked by popularity. The lowest rank is the most visited item.

RQ2: How Does the Hyperparameter α Impact Accurracy and CTR Results? In Table 3, we show the NDCG@3 for each of the settings of α in our offline tests and the lift in CTR during the online tests.

In the offline experiments increasing the α hyperparameter beyond 0.2 leads to a decrease in performance. As less popular items are recommended, accuracy suffers. Online we find a similar result, higher values of α do not correlate with a higher CTR. However, maximal online performance is reached with the control setting of $\alpha = 0.5$.

So, while a higher α results in a higher coverage and a lower Gini coefficient, both the click-through rate and the NDCG show a decrease in performance when we increase α too much. In our news use-cases, exposure equality and countering popularity bias need to be balanced with recommendation performance. Popular items are relevant to many users, and so if we want to showcase more, less popular, items, we might need to accept a performance decline.

Table 3. NDCG@3 (offline) results and CTR (online) results. CTR results are relative performance compared to the control setting ($\alpha = 0.5$).

α	NDCG@3 (%)					CTR lift (%)				
	0.0	0.2	0.5	0.7	1.0	0.0	0.2	0.5	0.7	1.0
NP1	8.52	**9.15**	7.68	5.27	0.70	−6.40	−4.05	**0**	−4.82	−21.26
NP2	5.54	**6.43**	6.41	4.47	1.07	−3.28	−1.28	**0**	−6.12	−26.45
NP3	6.44	**7.02**	6.48	4.16	0.40	−6.87	−3.91	**0**	−6.93	−31.60

RQ3: Do the Offline and Online Results Agree? In the offline results, the optimal setting for all datasets is $\alpha = 0.2$. However, in our online results, $\alpha = 0.2$ is not optimal, instead $\alpha = 0.5$ is the optimal setting.

Our datasets, like many news datasets, show an unbalanced reading behaviour, indicated by the high Gini coefficient in Table 1. Users read the most popular items much more often than the other items. This popularity bias leads to higher performance in offline results for algorithms with more popularity bias (lower α). However, in the production setting, recommending mostly popular items leads to recommending popular items not related to the context item. Users looking for related articles do not click on these popularity-based recommendations. These results follow the common finding, due to popularity bias offline and online results do not align nicely. However, we can see the value of the offline experimentation in the performance of the $\alpha = 1$ setting. The bad offline performance is reflected in the online results.

5 Conclusion

We find that while the hyperparameter α is able to counteract popularity bias, it is only a proxy for true exposure equality. Therefore, further research is required on how to combat the popularity bias of the ItemKNN algorithm. Secondly, we note that our offline and online results do not align, due to the inherent popularity bias in typical offline evaluation [4,5,23]. Our findings suggest that it is worthwhile to opt for suboptimal offline test results in terms of accuracy, but with a lower Gini index. However, a trade-off should be sought between fair exposure and user experience. We leave a thorough investigation of this trade-off and a framework for determining the setting most likely to perform best in online tests for future work. Finally, we note that our results are limited to the news domain. We see no reason to believe that our findings will not generalize to other domains, as they were not dependent on specific characteristics of the news context. However, it is our aim to replicate these findings in other domains, provided we find partners to perform these trials with.

References

1. Abdollahpouri, H., Burke, R., Mobasher, B.: Controlling popularity bias in learning-to-rank recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, pp. 42–46. Association for Computing Machinery, New York (2017). https://doi.org/10.1145/3109859.3109912. ISBN 9781450346528
2. Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B.: The unfairness of popularity bias in recommendation. In: CEUR Workshop Proceedings, vol. 2440 (2019). https://ceur-ws.org/Vol-2440/paper4.pdf
3. Bambini, R., Cremonesi, P., Turrin, R.: A recommender system for an IPTV service provider: a real large-scale production environment. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 299–331. Springer, Boston (2011). https://doi.org/10.1007/978-0-387-85820-3_9
4. Beel, J., Genzmehr, M., Langer, S., Nürnberger, A., Gipp, B.: A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation. In: Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, RepSys 2013, pp. 7–14. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2532508.2532511. ISBN 9781450324656
5. Beel, J., Langer, S.: A comparison of offline evaluations, online evaluations and user studies in the context of research-paper recommender systems. In: Kapidakis, S., Mazurek, C., Werla, M. (eds.) Research and Advanced Technology for Digital Libraries, pp. 153–168. Springer Cham (2015). https://doi.org/10.1007/978-3-319-24592-8_12. ISBN 978-3-319-24592-8
6. Castells, P., Hurley, N., Vargas, S.: Novelty and diversity in recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 603–646. Springer, New York (2022). https://doi.org/10.1007/978-1-0716-2197-4_16. ISBN 978-1-0716-2197-4
7. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. **22**(1), 143–177 (2004). https://doi.org/10.1145/963770.963776. ISSN 1046–8188
8. Eksombatchai, C., et al.: Pixie: a system for recommending 3+ billion items to 200+ million users in real-time. In: Proceedings of the 2018 World Wide Web Conference, WWW 2018, pp. 1775–1784. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). https://doi.org/10.1145/3178876.3186183. ISBN 9781450356398
9. Ferrari Dacrema, M., Boglio, S., Cremonesi, P., Jannach, D.: A troubling analysis of reproducibility and progress in recommender systems research. ACM Trans. Inf. Syst. **39**(2) (2021). https://doi.org/10.1145/3434185. ISSN 1046–8188
10. Ferrari Dacrema, M., Cremonesi, P., Jannach, D.: Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender System, RecSys 2019, pp. 101–109. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3298689.3347058. ISBN 9781450362436

11. Garcin, F., Faltings, B., Donatsch, O., Alazzawi, A., Bruttin, C., Huber, A.: Offline and online evaluation of news recommender systems at Swissinfo.ch. In: Proceedings of the 8th ACM Conference on Recommender Systems, RecSys 2014, pp. 169–176. Association for Computing Machinery, New York (2014). https://doi.org/10.1145/2645710.2645745. ISBN 9781450326681
12. Gruson, A., et al.: Offline evaluation to make decisions about playlist recommendation algorithms. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, pp. 420–428. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3289600.3291027. ISBN 9781450359405
13. Gunawardana, A., Shani, G., Yogev, S.: Evaluating recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 547–601. Springer, New York (2022). https://doi.org/10.1007/978-1-0716-2197-4_15 ISBN 978-1-0716-2197-4
14. Järvelin, K., Kekäläinen, J.: cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. **20**(4), 422–446 (2002). https://doi.org/10.1145/582415.582418. ISSN 1046–8188
15. Kersbergen, B., Sprangers, O., Schelter, S.: Serenade - low-latency session-based recommendation in e-commerce at scale. In: Proceedings of the 2022 International Conference on Management of Data, SIGMOD 2022, pp. 150–159. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3514221.3517901. ISBN 9781450392495
16. Ludewig, M., Mauro, N., Latifi, S., Jannach, D.: Performance comparison of neural and non-neural approaches to session-based recommendation. In: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, pp. 462–466. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3298689.3347041. ISBN 9781450362436
17. Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B., Burke, R.: Feedback loop and bias amplification in recommender systems. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM 2020, pp. 2145–2148. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3340531.3412152. ISBN 9781450368599
18. Mei, M.J., Zuber, C., Khazaeni, Y.: A lightweight transformer for next-item product recommendation. In: Proceedings of the 16th ACM Conference on Recommender Systems, RecSys 2022, pp. 546–549. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3523227.3547491. . ISBN 9781450392785
19. Pellegrini, R., Zhao, W., Murray, I.: Don't recommend the obvious: estimate probability ratios. In: Proceedings of the 16th ACM Conference on Recommender Systems, RecSys 2022, pp. 188–197. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3523227.3546753. ISBN 9781450392785
20. Rehorek, T., Biza, O., Bartyzal, R., Kordik, P., Povalyev, I., Podstavek, O.: Comparing offline and online evaluation results of recommender systems. In: REVEAL 2018: Proceedings of the Workshop on Offline Evaluation for Recommender Systems (2018). https://users.fit.cvut.cz/rehorto2/files/comparing-offline-online.pdf
21. Rossetti, M., Stella, F., Zanker, M.: Contrasting offline and online results when evaluating recommendation algorithms. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016, pp. 31–34. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2959100.2959176. ISBN 9781450340359

22. Verachtert, R., Michiels, L., Goethals, B.: Are we forgetting something? Correctly evaluate a recommender system with an optimal training window. In: Proceedings of the Perspectives on the Evaluation of Recommender Systems Workshop 2022, Seattle, WA, USA. CEUR-WS.org, September 2022
23. Zangerle, E., Bauer, C.: Evaluating recommender systems: survey and framework. ACM Comput. Surv. **55**(8) (2022). https://doi.org/10.1145/3556536. ISSN 0360–0300
24. Zhu, Z., He, Y., Zhao, X., Caverlee, J.: Popularity bias in dynamic recommendation. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2021, pp. 2439–2449. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3447548.3467376. ISBN 9781450383325